COMPUTER HANDBOOK

HUSKEY AND KORN

510.784
H959
c.4

COMPUTER HANDBOOK

McGRAW-HILL HANDBOOKS

Abbott and Stetka · National Electrical Code Handbook, 10th ed.
Aljian · Purchasing Handbook
American Institute of Physics · American Institute of Physics Handbook
American Society of Mechanical Engineers · ASME Handbooks:
Engineering Tables
Metals Engineering—Design
Metals Engineering—Processes
Metals Properties
American Society of Tool and Manufacturing Engineers:
Die Design Handbook
Handbook of Fixture Design
Tool Engineers Handbook, 2d ed.
Beeman · Industrial Power Systems Handbook
Berry, Bollay, and Beers · Handbook of Meteorology
Blatz · Radiation Hygiene Handbook
Brady · Materials Handbook, 8th ed.
Burington · Handbook of Mathematical Tables and Formulas, 3d ed.
Burington and May · Handbook of Probability and Statistics with Tables
Carroll · Industrial Instrument Servicing Handbook
Cockrell · Industrial Electronics Handbook
Condon and Odishaw · Handbook of Physics
Considine · Process Instruments and Controls Handbook
Crocker · Piping Handbook, 4th ed.
Croft and Carr · American Electricians' Handbook, 8th ed.
Davis · Handbook of Applied Hydraulics, 2d ed.
Dudley · Gear Handbook
Etherington · Nuclear Engineering Handbook
Factory Mutual Engineering Division · Handbook of Industrial Loss Prevention
Fink · Television Engineering Handbook
Frick · Petroleum Production Handbook, 2 vols.
Guthrie · Petroleum Products Handbook
Harris · Handbook of Noise Control
Harris and Crede · Shock and Vibration Handbook, 3 vols.
Henney · Radio Engineering Handbook, 5th ed.
Hunter · Handbook of Semiconductor Electronics, 2d ed.
Huskey and Korn · Computer Handbook
Jasik · Antenna Engineering Handbook
Johnson and Auth · Fuels and Combustion Handbook
Juran · Quality-control Handbook
Kallen · Handbook of Instrumentation and Controls
Ketchum · Structural Engineers' Handbook, 3d ed.
King · Handbook of Hydraulics, 4th ed.
Knowlton · Standard Handbook for Electrical Engineers, 9th ed.
Koelle · Handbook of Astronautical Engineering
Korn and Korn · Mathematical Handbook for Scientists and Engineers
Kurtz · The Lineman's Handbook, 3d ed.
La Londe and Janes · Concrete Engineering Handbook
Landee, Davis, and Albrecht · Electronic Designers' Handbook
Lange · Handbook of Chemistry, 10th ed.
Laughner and Hargan · Handbook of Fastening and Joining of Metal Parts
Le Grand · The New American Machinist's Handbook
Liddell · Handbook of Nonferrous Metallurgy, 2d ed.
Magill, Holden, and Ackley · Air Pollution Handbook
Manas · National Plumbing Code Handbook
Mantell · Engineering Materials Handbook
Marks and Baumeister · Mechanical Engineers' Handbook, 6th ed.
Markus · Handbook of Electronic Control Circuits
Markus and Zeluff · Handbook of Industrial Electronic Circuits
Markus and Zeluff · Handbook of Industrial Electronic Control Circuits
Maynard · Industrial Engineering Handbook
Merritt · Building Construction Handbook
Moody · Petroleum Exploration Handbook
Morrow · Maintenance Engineering Handbook
Perry · Chemical Business Handbook
Perry · Chemical Engineers' Handbook, 3d ed.
Shand · Glass Engineering Handbook, 2d ed.
Staniar · Plant Engineering Handbook, 2d ed.
Streeter · Handbook of Fluid Dynamics
Stubbs · Handbook of Heavy Construction
Terman · Radio Engineers' Handbook
Truxal · Control Engineers' Handbook
Urquhart · Civil Engineering Handbook, 4th ed.
Walker · NAB Engineering Handbook, 5th ed.
Woods · Highway Engineering Handbook
Yoder, Heneman, Turnbull, and Stone · Handbook of Personnel Management and Labor Relations

COMPUTER HANDBOOK

Edited by

HARRY D. HUSKEY, Ph.D.

Professor of Electrical Engineering and Mathematics
University of California
Berkeley, Calif.

and

GRANINO A. KORN, Ph.D.

Professor of Electrical Engineering
University of Arizona
Tucson, Ariz.

FIRST EDITION

McGRAW-HILL BOOK COMPANY, INC.

New York Toronto London

1962

COMPUTER HANDBOOK

Copyright © 1962 by the McGraw-Hill Book Company, Inc. Printed in the United States of America. All rights reserved. This book, or parts thereof, may not be reproduced in any form without permission of the publishers.

Section 12.7 ("Magnetic-core Storage and Switching Techniques") was copyrighted in 1959 by Ampex Corporation.

Library of Congress Catalog Card Number: 60-15286

31477

THE MAPLE PRESS COMPANY, YORK, PA.

CONTRIBUTORS

CHARLES W. ADAMS, President, Charles W. Adams Associates, Inc., Bedford, Mass. (*Section 21, Applications of Digital Computers*)

ERNEST G. ANDREWS, Manager, Preliminary Design, Sanders Associates, Nashua, N.H. (*Section 20, General-purpose Computers*)

WILLIAM R. ARSENAULT, Vice-President, FMA, Inc., El Segundo, Calif. (*Section 20, General-purpose Computers*)

MORTON M. ASTRAHAN, Systems Engineer, International Business Machines Corporation, Advanced Systems Development Division Laboratory, San Jose, Calif. (*Section 18, Input and Output*)

ISAAC L. AUERBACH, President and Technical Director, Auerbach Electronics Corporation, Philadelphia, Pa. (*Section 10, Components. Section 12, Memory Devices*)

LOUIS BAUER, Director, Project CYCLONE, Reeves Instrument Corporation, Garden City, N.Y. (*Section 5, Electronic Analog Computers: Significant Applications*)

EDWARD M. BILLINGHURST, Project Engineer, Beckman/Berkeley Division, Richmond, Calif. (*Section 2, Electronic Analog Computers: Coefficient Potentiometers, Operational Amplifiers, and Networks*)

WERNER BUCHHOLZ, Senior Systems Planner, International Business Machines Corporation, Poughkeepsie, N.Y. (*Section 16, Digital-computer-system Design*)

DUDLEY A. BUCK,* Formerly with Massachusetts Institute of Technology, Cambridge, Mass. (*Section 13, Switching Circuits*)

ARTHUR W. BURKS, Professor, Department of Philosophy, University of Michigan, Ann Arbor, Mich. (*Section 14, Information Coding and Switching Theory*)

RICHARD G. CANNING, Consultant, Vista, Calif. (*Section 21, Applications of Digital Computers*)

VICTOR B. COREY, President, Palomar Scientific Corporation, Palo Alto, Calif., and Vice-President, United Control Corporation, Seattle, Wash.; formerly Technical Director, Donner Scientific Company, Concord, Calif. (*Section 5, Electronic Analog Computers: Significant Applications*)

* Deceased

WILLIAM J. DIXON, President, Computer Engineering Associates, Inc., Pasadena, Calif. (*Section 9, Network-type Direct-analogy Computers and Field-problem Analogies*)

J. JAMES EBERS,* Formerly with Bell Telephone Laboratories, Inc., Allentown Laboratory. (*Section 10, Components*)

HAROLD L. EHLERS, Supervisor, Flight Control Section, Autonetics Division, North American Aviation, Inc., Downey, Calif. (*Section 4, Electronic Analog Computers: Control Circuits, Computer Operation, and System Design. Section 7, Transistorized Electronic Analog Computers*)

M. L. EMBREE, Member of Technical Staff, Laureldale Laboratory, Bell Telephone Laboratories, Inc., Laureldale, Pa. (*Section 10, Components*)

WILLIAM A. FARRAND, Staff of Chief Engineer, Autonetics Division, North American Aviation, Inc., Downey, Calif. (*Section 18, Input and Output*)

JEROME L. GREENSTEIN, Senior Research Engineer, General Dynamics/Astronautics, San Diego, Calif. (*Section 6, Electronic Analog Computers: Special Components and Techniques*)

DONALD T. GREENWOOD, Associate Professor of Instrumentation Engineering, University of Michigan, Ann Arbor, Mich. (*Section 9, Network-type Direct-analogy Computers and Field-problem Analogies*)

WILLIAM F. GUNNING, Vice-President—Technical Director, Epsco-West, Inc., Anaheim, Calif. (*Section 16, Digital-computer-system Design*)

RICHARD W. HAMMING, Member Technical Staff, Bell Telephone Laboratories, Inc., Murray Hill, N.J. (*Section 14, Information Coding and Switching Theory*)

ALBERT S. HOAGLAND, Senior Engineer, International Business Machines Research Laboratory, San Jose, Calif. (*Section 12, Memory Devices*)

WALTER HOCHWALD, Supervisor, Inertial Navigation Department, Autonetics Division, North American Aviation, Inc., Downey, Calif. (*Section 4, Electronic Analog Computers: Control Circuits, Computer Operation, and System Design. Section 7, Transistorized Electronic Analog Computers*)

ARTHUR W. HOLT, Chief Electronic Engineer, Rabinow Engineering Company, Washington, D.C. (*Section 12, Memory Devices*)

R. M. HOWE, Professor of Aeronautical Engineering, University of Michigan, Ann Arbor, Mich. (*Section 5, Electronic Analog Computers: Significant Applications*)

HARRY D. HUSKEY, Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif. (*Section 10, Components. Section 11, Single-input Component Circuits. Section 12, Memory Devices. Section 13, Switching Circuits. Section 15, Digital-computer Arithmetic. Section 16,*

* Deceased

Digital-computer-system Design. Section 17, Introduction to Coding and Problem Logic. Section 20, General-purpose Computers. Section 21, Applications of Digital Computers)

ARVID W. JACOBSON, Professor, Department of Mathematics, Wayne State University, Detroit, Mich. (*Section 21, Applications of Digital Computers*)

E. CALVIN JOHNSON, Head, Computer Development Department, Research Laboratories Division, The Bendix Corporation, Detroit, Mich. (*Section 21, Applications of Digital Computers*)

ROBERT R. JOHNSON, Manager—Engineering, Computer Department, General Electric Company, Phoenix, Ariz. (*Section 19, Special-purpose Computers*)

WALTER J. KARPLUS, Associate Professor of Engineering, University of California, Los Angeles, Calif. (*Section 9, Network-type Direct-analogy Computers and Field-problem Analogies*)

JEROME D. KENNEDY, Sr., Vice-President, Applied Dynamics, Inc., Ann Arbor, Mich. (*Section 6, Electronic Analog Computers: Special Components and Techniques*)

GRANINO A. KORN, Professor of Electrical Engineering, University of Arizona, Tucson, Ariz. (*Section 1, Introduction and Block-diagram Notation. Section 2, Electronic Analog Computers: Coefficient Potentiometers, Operational Amplifiers, and Networks. Section 3, Electronic Analog Computers: Multipliers and Function Generators. Section 4, Electronic Analog Computers: Control Circuits, Computer Operation, and System Design. Section 5, Electronic Analog Computers: Significant Applications. Section 6, Electronic Analog Computers: Special Components and Techniques. Section 8, Miscellaneous Mechanical and Electrical Analog-computing Systems*)

THERESA M. KORN. (*Section 2, Electronic Analog Computers: Coefficient Potentiometers, Operational Amplifiers, and Networks. Section 5, Electronic Analog Computers: Significant Applications*)

L. D. KOVACH, Design Specialist, Engineering Computer Group, Douglas Aircraft Company, El Segundo Division, and Professor of Mathematics, Pepperdine College. (*Section 5, Electronic Analog Computers: Significant Applications. Section 6, Electronic Analog Computers: Special Components and Techniques*)

THADDEUS J. KUSTO, Director of Research, Computer Systems, Inc., New York, N.Y. (*Section 4, Electronic Analog Computers: Control Circuits, Computer Operation, and System Design*)

R. M. LEGER, Chief, Flight Performance and Analysis, General Dynamics/Astronautics, San Diego, Calif. (*Section 6, Electronic Analog Computers: Special Components and Techniques*)

BRAM J. LOOPSTRA, N.V. Electrologica, Amsterdam, The Netherlands. (*Section 11, Single-input Component Circuits*)

BERNARD D. LOVEMAN, Project Engineer, Project CYCLONE, Reeves Instrument Corporation, Garden City, N.Y. (*Section 2, Electronic Analog Computers: Coefficient Potentiometers, Operational Amplifiers, and Networks. Section 3, Electronic Analog Computers: Multipliers and Function Generators. Section 4, Electronic Analog Computers: Control Circuits, Computer Operation, and System Design*)

LOWELL S. MICHELS, Principal Engineer, Bendix-Pacific Division, The Bendix Corporation, North Hollywood, Calif. (*Section 18, Input and Output*)

W. K. McGREGOR, ARO, Inc., Tullahoma, Tenn. (*Section 6, Electronic Analog Computers: Special Components and Techniques*)

JOHN McLEOD, Design Specialist, General Dynamics/Astronautics, San Diego, Calif. (*Section 5, Electronic Analog Computers: Significant Applications*)

H. F. MEISSINGER, Assistant Manager, Systems Development Group, Hughes Aircraft Company, Culver City, Calif. (*Section 5, Electronic Analog Computers: Significant Applications*)

CHARLES D. MORRILL, Manager, Computer, Guidance, Simulator, and Test Equipment Engineering, Goodyear Aircraft Corporation, Akron, Ohio. (*Section 3, Electronic Analog Computers: Multipliers and Function Generators*)

E. MORRISON, Assistant Director, Applied Research Laboratory, University of Arizona, Tucson, Ariz. (*Section 5, Electronic Analog Computers: Significant Applications*)

MAX PALEVSKY, President, Packard-Bell Computer Company, Los Angeles, Calif. (*Section 19, Special-purpose Computers*)

CHARLES F. PULVÁRI, Professor, Department of Electrical Engineering, The Catholic University of America, Washington, D.C. (*Section 12, Memory Devices*)

VINCENT C. RIDEOUT, Professor of Electrical Engineering, University of Wisconsin, Madison, Wis. (*Section 5, Electronic Analog Computers: Significant Applications*)

STANLEY ROGERS, Head, Data Systems, General Dynamics/San Diego, San Diego, Calif. (*Section 4, Electronic Analog Computers: Control Circuits, Computer Operation, and System Design*)

SAUL ROSEN, Consultant, Computing and Programming Systems, Philadelphia, Pa. (*Section 21, Applications of Digital Computers*)

ARTHUR I. RUBIN, Director, Princeton Computation Center, Electronic Associates, Inc., Princeton, N.J. (*Section 5, Electronic Analog Computers: Significant Applications*)

MORRIS RUBINOFF, Professor, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pa. (*Section 21, Applications of Digital Computers*)

PAUL E. RUSSELL, Professor and Head, Department of Electrical Engineering, University of Arizona, Tucson, Ariz. (*Section 6, Electronic Analog Computers: Special Components and Techniques*)

PAUL SAVET, Staff Scientist, Arma Division, American Bosch Arma Corporation, Garden City, N.Y. (*Section 8, Miscellaneous Mechanical and Electrical Analog-computing Systems*)

HERMANN SCHMID, Senior Design Engineer, Link Division of General Precision, Inc., Binghamton, N.Y. (*Section 7, Transistorized Electronic Analog Computers*)

CHARLES H. SINGLE, Manager, Computer Engineering, Beckman/Berkeley Division, Richmond, Calif. (*Section 2, Electronic Analog Computers: Coefficient Potentiometers, Operational Amplifiers, and Networks*)

ROGER SISSON, Aeronutronic, Ford Motor Company, Newport Beach, Calif. (*Section 21, Applications of Digital Computers*)

DAVID SLEPIAN, Member Technical Staff, Bell Telephone Laboratories, Inc., Murray Hill, N.J. (*Section 14, Information Coding and Switching Theory*)

WALTER W. SOROKA, Professor of Mechanical Engineering, University of California, Berkeley, Calif. (*Section 8, Miscellaneous Mechanical and Electrical Analog-computing Systems. Section 9, Network-type Direct-analogy Computers and Field-problem Analogies*)

RAYMOND STUART-WILLIAMS, Technical Director, Ampex Computer Products Company, Ampex Corporation, Culver City, Calif. (*Section 12, Memory Devices*)

RAGNAR THORENSEN, Director, Magnavox Research Laboratories, Torrance, Calif. (*Section 16, Digital-computer-system Design*)

R. P. WASHBURN, Analyst, Marketing Research, Computer Department, General Electric Co., Phoenix, Ariz. (*Section 5, Electronic Analog Computers: Significant Applications. Section 9, Network-type Direct-analogy Computers and Field-problem Analogies*)

JAMES H. WILKINSON, Mathematics Division, National Physical Laboratory, Teddington, Middlesex, England. (*Section 21, Applications of Digital Computers*)

FREDERIC C. WILLIAMS, Professor, Department of Electrical Engineering, The University, Manchester, England. (*Section 12, Memory Devices*)

MICHAEL WOODGER, Mathematics Division, National Physical Laboratory, Teddington, Middlesex, England. (*Section 17, Introduction to Coding and Problem Logic*)

CHARLES W. WORLEY, Consulting Engineer (formerly Market Development Manager, Electronic Associates, Inc., Long Branch, N.J.). (*Section 5, Electronic Analog Computers: Significant Applications*)

PREFACE

The *Computer Handbook* in one comprehensive volume presents the general principles of design and utilization of both analog and digital computers. In either field sufficient detail is presented so that anyone competent in the field can proceed to construct a computer, or, having a computer, can proceed to use it. Components are discussed in detail, and many actual circuit diagrams have been included as concrete examples of design principles or for direct adaptation to the designer's problem. The organization of problems for solution on computers is described, and examples of applications of computers to a variety of problems are given.

For the younger engineer and for newcomers to the control and computer fields, the handbook provides quick access to exactly that industrial know-how which is necessarily neglected in a modern engineering-school curriculum stressing principles rather than techniques. For those desiring to solve problems on computers, techniques for producing flow diagrams are described and representative problems are worked out.

Since analog-computer design is often intimately related to computer applications and problem-solving methods, a rather comprehensive review of analog-computer applications and methodology has been added. A very large number of special computer setups and trick circuits are presented in tables and grouped illustrations for convenient reference.

Sections 1 through 9 are concerned with analog computers, while Sections 10 through 21 cover digital computers.

Specifically, Section 1 introduces the *basic terminology* for analog computers, and Sections 2 and 3 describe *the design of electronic-analog-computer building blocks,* including late *electronic multiplier and resolver circuits.* Section 4 deals with the *design of analog-computer systems, problem checking, computer-laboratory organization,* and *computer maintenance.*

Section 5 reviews the significant applications of electronic analog computers. *Control-system design, flight simulation, partial system tests,* and *process-control applications* are described; the section also contains information on *random-process studies, algebraic-equation solvers, linear programming,* and *solution of partial differential equations* with electronic analog computers. *Economic dispatch computers* are described as examples of special-purpose machines.

Section 6 introduces newer techniques, such as *dynamic-storage computation, repetitive computer techniques for statistical problems,* and *combined*

analog-digital computation. A special part deals with the use of *analog computers as control-system components.* Section 7 is entirely devoted to *solid-state* (transistor) *analog-computer components* and describes many new solid-state circuits.

Sections 8 and 9 deal entirely with important analog techniques less familiar to many engineers, viz., *network-type analogies for fields, structures,* and *power systems,* and with *mechanical, electromechanical, hydrodynamic,* and *heat-transfer computing elements.*

Section 10 presents the details of *components* used in digital computers, with particular emphasis on *semiconductor diodes* and *transistors.* Section 11 deals with *cathode* and *emitter followers, signal modifying circuits,* and *amplifiers. Ferromagnetic shift registers, magnetic drum* and *magnetic tape systems,* and *magnetic core memories* are discussed in detail in Section 12.

Section 13 describes *logical circuits,* including AND, OR, and other *switching circuits.* Section 14 presents material on *error-detecting* and *error-correcting codes* and on the use of *Boolean algebra* in switching-circuit design. *Arithmetic circuits* are discussed in Section 15. Section 16 on digital-computer-system design considers *plugboard control* as well as *stored-program control,* and compares *synchronous* and *asynchronous* aspects of design. *Programming* and *coding* and a description of the algorithmic language *ALGOL* is given in Section 17. Section 18 on input and output discusses *paper-tape* and *punched-card devices, symbol-display devices, analog-conversion devices,* and *plotters.*

Section 19 discusses various *special-purpose computers* including the *digital differential analyzer.* Section 20 presents material on *general-purpose computers,* and Section 21 discusses *scientific* and *engineering applications, data-processing applications, management applications, simulation,* and *process control* by use of computers.

This handbook is a result of interest originally expressed by the Professional Group on Electronic Computers of the Institute of Radio Engineers. Jean Felker of Bell Telephone Laboratories was helpful in the initial arrangements.

It has been the editors' privilege to work with a really outstanding group of contributors, including both computer design experts and applications specialists. In addition to the contributors proper, the editors are very grateful to many individuals and organizations for contributing engineering data, circuits, and general information. The following persons deserve particular mention:

S. I. Abraham	General Dynamics/Fort Worth
J. P. Barloe	International Business Machines Corp.
R. M. Beck	Packard-Bell Electronics Corp.
C. G. Blanyer	Hydel, Inc.

J. S. Bowen	Aircraft-Marine Products
D. R. Brown	Massachusetts Institute of Technology
J. C. Chu	Sperry-Rand Corp.
W. Comley	Douglas Aircraft Company
M. Connelly	Massachusetts Institute of Technology
F. W. Davis	General Dynamics/Fort Worth
C. M. Edwards	Bendix Corporation
D. C. Evans	Bendix Corporation
N. Fuller	General Dynamics/San Diego
M. H. Gerber	Computer Systems, Inc.
H. R. Grosch	Consultant
L. E. Heizer	General Dynamics/Fort Worth
A. W. Hertzog	U.S. Naval Training Devices Center
J. L. Hussey	Beckman Instruments, Inc.
P. H. Kinsey	General Electric Co.
L. Lofgren	Swedish Institute of National Defense
H. Low	Space Technology Laboratories
R. H. Malter	G. A. Philbrick Researches, Inc.
R. D. McCoy	Reeves Instrument Corporation
H. Mori	Hydel, Inc.
Mrs. Mercedes O'Dea	Electronic Design
R. W. Olmstead	Electronic Associates, Inc.
H. Paynter	Massachusetts Institute of Technology
P. Rafield	Reeves Instrument Corporation
G. Risberg	Swedish Institute of National Defense
R. F. Shanahan	General Electric Co.
D. H. Sheingold	G. A. Philbrick Researches, Inc.
C. V. L. Smith	Aberdeen Proving Ground
R. Stern	Computer Systems, Inc.
E. Sterne	Beckman Instruments, Inc.
G. S. Stubbs	Franklin Institute Laboratories
N. H. Taylor	Massachusetts Institute of Technology
P. Thaulow	Beckman Instruments, Inc.
D. M. Trotsky	GPS Instrument Co.
H. S. Wolanski	General Dynamics/Fort Worth
A. Wolcott	International Business Machines Corp.

Credit also goes to Velma R. Huskey for assisting with the editing, proofing, and indexing of the digital part, and to the Bendix Corporation for secretarial assistance.

G. A. Korn

H. D. Huskey

CONTENTS

Contributors vii

Preface xiii

ANALOG COMPUTERS

Section 1. Introduction and Block-diagram Notation 1–3
Section 2. Electronic Analog Computers: Coefficient Potentiometers, Operational Amplifiers, and Networks 2–1
Section 3. Electronic Analog Computers: Multipliers and Function Generators 3–1
Section 4. Electronic Analog Computers: Control Circuits, Computer Operation, and System Design 4–1
Section 5. Electronic Analog Computers: Significant Applications . . 5–1
Section 6. Electronic Analog Computers: Special Components and Techniques 6–1
Section 7. Transistorized Electronic Analog Computers 7–1
Section 8. Miscellaneous Mechanical and Electrical Analog-computing Systems 8–1
Section 9. Network-type Direct-analogy Computers and Field-problem Analogies 9–1

DIGITAL COMPUTERS

Section 10. Components 10–3
Section 11. Single-input Component Circuits 11–1
Section 12. Memory Devices 12–1
Section 13. Switching Circuits 13–1
Section 14. Information Coding and Switching Theory 14–1
Section 15. Digital-computer Arithmetic 15–1
Section 16. Digital-computer-system Design 16–1
Section 17. Introduction to Coding and Problem Logic 17–1

Section 18. Input and Output 18–1
Section 19. Special-purpose Computers 19–1
Section 20. General-purpose Computers 20–1
Section 21. Applications of Digital Computers 21–1

Index follows Section 21.

COMPUTER HANDBOOK

ANALOG COMPUTERS

Section 1

INTRODUCTION AND BLOCK-DIAGRAM NOTATION

By

GRANINO A. KORN, *Professor of Electrical Engineering, University of Arizona, Tucson, Ariz.*

CONTENTS

PART 1: INTRODUCTION

1.1.1. Analog Computers..... 1–3
1.1.2. An Appraisal of Analog Computers..... 1–4
1.1.3. Applications and Special Techniques..... 1–5

PART 2: BLOCK-DIAGRAM SYMBOLS FOR ELECTRONIC ANALOG COMPUTERS

1.2.1. Introduction..... 1–6
1.2.2. Suggested Principles of Notation..... 1–6
1.2.3. Standard Abbreviations..... 1–6

PART 1: INTRODUCTION

1.1.1. Analog Computers. In the most general sense, an *analog computer* is any physical system which establishes definite prescribed relations between continuously variable physical quantities. Definite relationships are necessarily stated in terms of mathematical relations. It follows that every analog computer, whether used for problem solving or for direct control, implements a requirement for at least approximate physical realization of a mathematical model, which can be understood and manipulated with relative ease.

Most analog-computing systems are combinations of *analog-computing elements* which establish elementary mathematical relations (e.g., addition, multiplication, integration). Analog computers can be classified according to the nature of the physical quantities used to represent mathematical variables and according to the computing elements used. Slide rules, network calculators, and Bush differential analyzers are, respectively, examples of mechanical, electrical, and electromechanical analog computers.

Analog computers used to solve problems represent each problem variable by a corresponding physical quantity (voltage, shaft displacement) on a convenient scale

(*problem scaling*, Sec. 5.1). These *machine variables* are made to obey mathematical relations corresponding to those of the given problem. In particular, a *differential analyzer* starts a set of machine variables from specified initial values and employs *analog integrators* (Secs. 2.2 and 8.1) together with other computing elements to enforce a system of prescribed ordinary differential equations; records of the machine variables constitute solutions of the differential equations (Sec. 5.1). Physical time serves, ultimately, as the independent variable in most such machines. The dependent variables are most frequently represented by instantaneous values of voltages (*electronic differential analyzer, d-c analog computer*) or shaft displacements (*electromechanical differential analyzer*, Sec. 8.1).

An analog *equation solver* determines solutions $x, y, \ldots$ of a system of simultaneous equations $f_1(x,y, \ldots) = 0$, $f_2(x,y, \ldots) = 0, \ldots$ either by rapid trial-and-error computations of an "error function" like $f_1^2(x,y, \ldots) + f_2^2(x,y, \ldots) + \ldots$ until the desired combination of unknowns $x, y, \ldots$ is found, or the machine enforces a system of ordinary differential equations whose solutions converge to the desired unknowns $x, y, \ldots$ (Sec. 5.11). Analogous methods determine a set of unknown parameters $x, y, \ldots$ which *maximize* or *minimize* a given function $F(x,y, \ldots)$ (Sec. 5.11).

1.1.2. An Appraisal of Analog Computers. Unlike digital computers, most analog computers employ distinct computing elements for each mathematical operation required to solve a given problem (parallel operation).[1] Parallel operation is an essential reason for the *very high computing speeds* possible with d-c analog computers: most problems, regardless of complication, are solved within 2 minutes by "slow" d-c analog computers and within fractions of a second by repetitive machines (Sec. 6.2). On the other hand, parallel operation imposes *practical limits on the complexity of problems* which can be solved on analog computers. Also, the cost of analog computers with component accuracies better than 0.1 per cent rises sharply for even small improvements in accuracy. The use of digital computers is thus indicated for computations requiring high accuracy.

The relatively low cost of some d-c analog computers is of interest in certain applications. Practical and useful low-cost d-c analog differential-equation solvers have been constructed and are particularly well suited for instruction purposes and for industrial organizations wishing to provide individual engineers with desk-top analog computers (Sec. 4.5). In the case of larger, more flexible, and more generally applicable d-c analog-computer installations, the gap between the costs of analog and digital computing equipment has closed; many research and development groups own d-c analog-computer installations whose cost easily rivals that of their large digital computers.

What are the concrete reasons for such investments in comparatively inaccurate computing equipment? In the writer's opinion, analog computers should *not* be regarded as low-cost substitutes for digital computers. Analog computers can only rarely compete with digital machines in the mass production of numerical data required as *end products* of scientific or engineering analyses. In such applications, the high computing speeds of analog equipment are often outweighed by the more advanced automatic programming of digital computers, even if only slide-rule accuracy is required.

The really important contributions of analog computers to modern research and development techniques go beyond mere numerical computation. In many applications, the analog approach functions as a direct aid to a research worker's or engineer's thinking process: an analog-computer setup serves as a *model* which helps to bridge the gap between mathematical symbolism and physical reality. Simple patchcord connections and potentiometer settings enable the system designer to create a scale model which permits convenient investigation of the performance and interaction of components and systems. New ideas can be tested at once: blocks of computing elements represent system components and are readily replaced or assembled into different system designs. Design parameters may be changed instantly by corresponding resistance changes, and high computing speeds permit optimization of system performance by successive parameter adjustments.[1]

The *time scale* of analog computations may be slowed down or speeded up within wide limits at the convenience of the experimenter. In particular, d-c analog-computing speeds are sufficiently high to permit representation of most dynamical and electromechanical processes on a 1:1 time scale (*simulation*).

With the computer running on a 1:1 time scale, actual physical components of a system under investigation can take the place of a corresponding block of computing elements (*partial system testing*). Such applications often yield very concrete design information: frequency, d-c analog-computer simulation can replace cumbersome breadboard testing and permits a design to progress directly to the prototype-model stage (Sec. 5.6).

Digital computers as well as analog computers can be set up as mathematical models representing physical processes. The inherent flexibility of digital-computer programming makes such machines particularly suitable for the representation of discrete random processes and problems involving logical decisions. While such applications of digital computers will become even more useful as computing speeds and memory capacities increase, the reformulation of problems for digital-computer programming is difficult for many present-day engineers. A significant reason for the remarkable successes of analog computers is their ready acceptance by engineering personnel.

1.1.3. Applications and Special Techniques. The best-known and most important applications of electronic analog computers involve the design of servomechanisms, aircraft stability and control, and process control (Secs. 5.3 to 5.9), but many other applications are of great interest (Secs. 5.2, 5.10, 5.11, and 6.6). In addition, essentially every electronic, electromechanical, or mechanical analog computing element has useful applications in control and instrumentation systems (Secs. 6.4, 8.1, and 8.2). Current trends in electronic analog computation point toward *increased computing speeds* through redesign of electronic computing elements and elimination of mechanical components; in addition, increased use of new and accurate *analog sampling, switching, and storage devices* will combine fast analog computation with a measure of decision-making and automatic-programming capability (Secs. 3.3, 4.2, 5.11, and 6.3). *Hybrid computation schemes* similarly combine analog and digital techniques to obtain favorable combinations of accuracy, computing speed, ease of programming, and/or cost (Secs. 6.3 and 6.5).

Electronic and direct-analog methods for solving *field problems* (partial differential equations) can often provide at least approximate solutions to extraordinarily large and complicated problems of this type (Secs. 5.10 and 9.1 to 9.5).[2] In particular, network-type direct-analog computers are especially well adapted to the very complex structural and vibration problems of the aircraft industry (Secs. 9.1 and 9.2).

References and Bibliography

1. Korn, G. A., and T. M. Korn: *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.
2. Karplus, W. J.: *Analog Simulation*, McGraw-Hill Book Company, Inc., New York, 1958.
3. Karplus, W. J., and W. W. Soroka: *Analog Methods*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1959.
4. Jackson, A. S.: *Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1960.
5. Johnson, C. L.: *Analog Computer Techniques*, McGraw-Hill Book Company, Inc., New York, 1956.
6. Smith, G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1959.
7. Wass, C. A.: *Introduction to Electronic Analogue Computers*, McGraw-Hill Book Company, Inc., New York, 1955.
8. Warfield, J. N.: *Electronic Analog Computers*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1959.

PART 2: BLOCK-DIAGRAM SYMBOLS FOR ELECTRONIC ANALOG COMPUTERS

1.2.1. Introduction. One does well to realize that analog-computer block diagrams serve at least two distinct purposes, viz.:

1. To illustrate and interpret the information flow through the analog model
2. To serve as instructions for making actual patching connections, and as records of such interconnections

While it would be fairly easy to agree on block-diagram symbols for purely illustrative purposes, symbols used to show actual interconnections will often exhibit differences of computers or patching arrangements designed by different manufacturers. Some computing elements (e.g., summers, integrators) are fairly well standardized, whereas others (servos, resolvers, special diode circuits) are not. Some such devices are not suited to rigid standardization, and engineers must familiarize themselves with a number of alternative symbols; there is not enough complication to cause real hardship.

1.2.2. Suggested Principles of Notation. *In all but the simplest block diagrams, all signal flow should be from left to right, so that each "line" of a block diagram can be read like the mathematical relation it represents.* Most feedback connections are thus not drawn in but can be picked up by their machine-variable symbol at the left of the diagram; the best patchbay designs are those duplicating this signal flow from left to right. Each "line" of the block diagram might represent a mathematical relation having some intuitive meaning (one equation of motion, the transformation of one coordinate, one input-output relation, etc.).

Block diagrams should show the identifying number of each computing element. It has become customary to write the identifying number (e.g., N, M in the figures) *inside* each block-diagram symbol, and other numerical data (amplifier gains, potentiometer settings) on the *outside*.

Where block diagrams are used for patching without setup sheets, it may be useful to indicate the destination of signal lines leaving the diagram on the right, with or without special symbols (Fig. 1.2.10).

A special notation problem arises where a machine variable (Z in Fig. 1.2.6*d*, X in 1.2.6*e*, and 1.2.7) drives a computer servomechanism or "master" electronic multiplier to produce two or more products for different "lines" of the block diagram. One may then

1. Show all these products generated at the output of the block-diagram "line" producing Z or X (Figs. 1.2.6*d* and 1.2.7*a*)
2. Show different servo-driven potentiometers or "slave" multipliers in different parts of the block diagram (Figs. 1.2.6*e* and 1.2.7*c*)

The second approach may be more convenient for setup purposes.

Fig. 1.2.11 shows a number of symbols for special diode and relay circuits which are available as packaged units in special computers. In general, the actual circuits would have to be shown.

Block-diagram symbols should be easy to draw but should not conflict with each other or with commonly accepted circuit symbols. Figure 1.2.12 shows three block-diagram symbols which tend to be misleading.

1.2.3. Standard Abbreviations. Table 1.2.1 shows proposed IRE standard abbreviations, which may be used with rectangular blocks instead of special symbols.

Table 1.2.1. Proposed IRE Standard Abbreviations

A	Amplifier
ADC	Analog-to-digital converter
DAC	Digital-to-analog converter
DVM	Digital voltmeter
EM	Electronic multiplier
EXT	External connections or trunk
FG	Function generator
IC	Initial condition
INT	Integrator
LIM	Limiter
P	Potentiometer
PB	Plotting board
REC	Recorder
REF	Reference voltage
SA	Servo amplifier
SM	Servomultiplier
SR	Servo resolver
RA	Relay amplifier
REL	Relay
SW	Switch
TD	Time delay

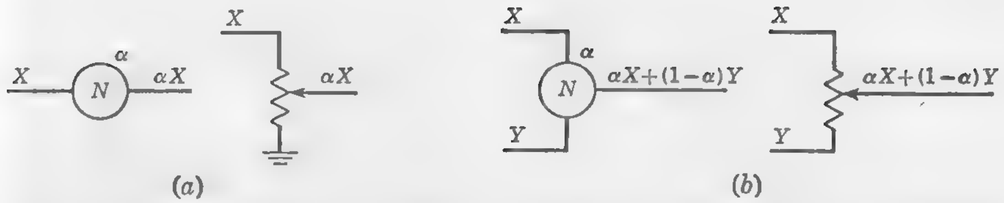


FIG. 1.2.1. *Proposed standard symbols for two-terminal coefficient-setting potentiometers* (*a*) and *three-terminal potentiometers* (*b*), with equivalent circuit diagrams. N is the identifying number of the potentiometer; α is the desired transfer function (including loading effects) and *not* necessarily the dial setting.

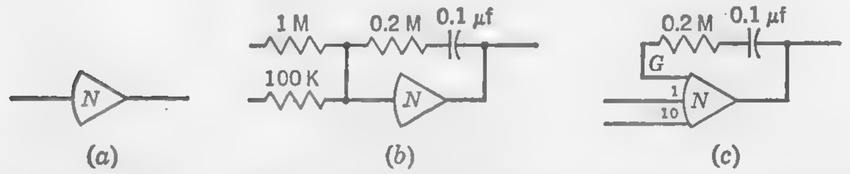


FIG. 1.2.2. *Proposed standard symbols for high-gain d-c amplifiers.* The notation of Fig. 1.1.2*c* is equivalent to that of Fig. 1.1.2*b* and is used with committed summing networks.

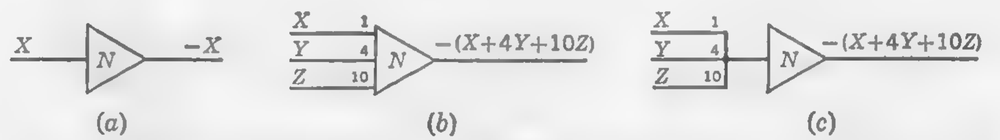


FIG. 1.2.3. *Proposed standard symbols for committed phase inverters* (*a*) and *summing amplifiers* (*b*). Input gains equal to 1 need not be shown. The notation of Fig. 1.2.3*c* can be used for summing amplifiers with many inputs.

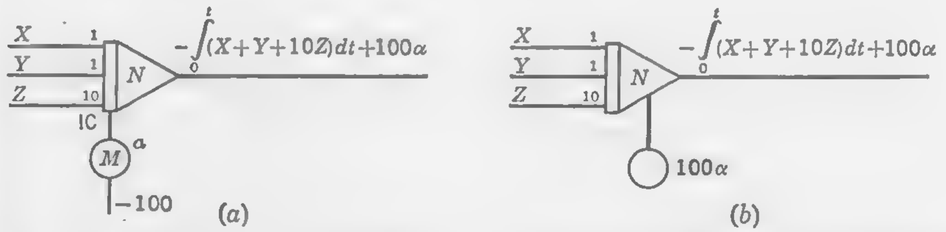
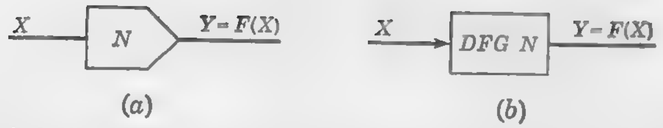


FIG. 1.2.4. Symbols for *committed integrators.* The symbol of Fig. 1.2.4*a* is for integrators with conventional relay resetting circuits; the reference-supply input to the initial-condition potentiometer can be replaced by the notation $-\alpha$ (or α for a +100-volt connection) if no patching connection is needed (**caution: some workers reverse the sign convention**). The symbol of Fig. 1.1.4*b* may be useful for integrators with built-in, floating, or electronic resetting circuits which do not involve phase inversion of the initial-condition voltage. Input gains equal to 1 need not be shown. A notation similar to Fig. 1.2.3*c* can be used for integrators with many inputs.



(a) (b)

FIG. 1.2.5. Symbols for *electronic function generators.* Figure 1.2.5*a* is the proposed IRE standard. The box may show a graph of the function generated; this is a useful short notation for special diode function generators (dead space, Coulomb friction, etc.). Note also that (1) diode function generators may have two or more inputs (X and $-X$; X and a reference voltage; or X_1, X_2) and (2) two or more diode function generators can be combined.

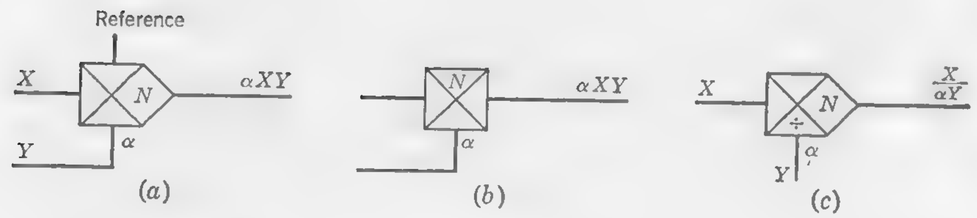
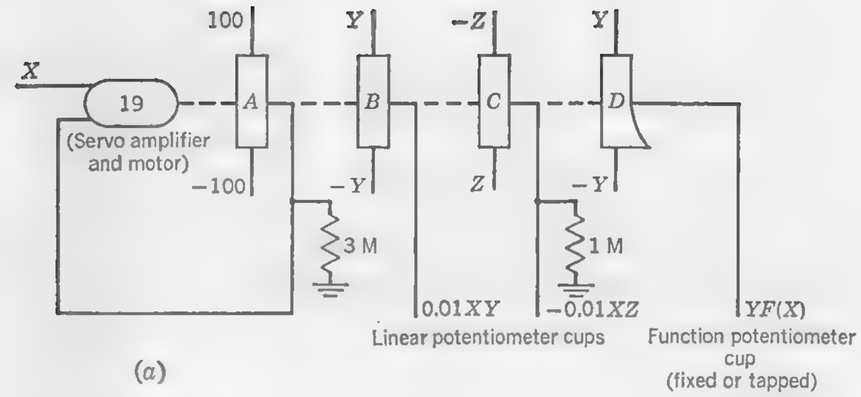


FIG. 1.2.6. Symbols for *electronic multipliers.* Figure 1.2.6*a* is the proposed IRE standard. The coefficient α can be positive or negative and might be omitted if $\alpha = 0.01$, i.e., if two 100-volt inputs produce a 100-volt output with correct sign. The reference connection will be shown only if it must be patched. Figure 1.2.6*c* is a possible symbol for a multiplier in a built-in division mode. Figure 1.2.6*d* and *e* show *slave operation of time-division multipliers:* channels are here labeled A, B, ... The notation of Fig. 1.2.6*e* permits one to separate the symbols for the two channels in the diagram. Some electronic multipliers have two X inputs (X and $-X$).



(a)

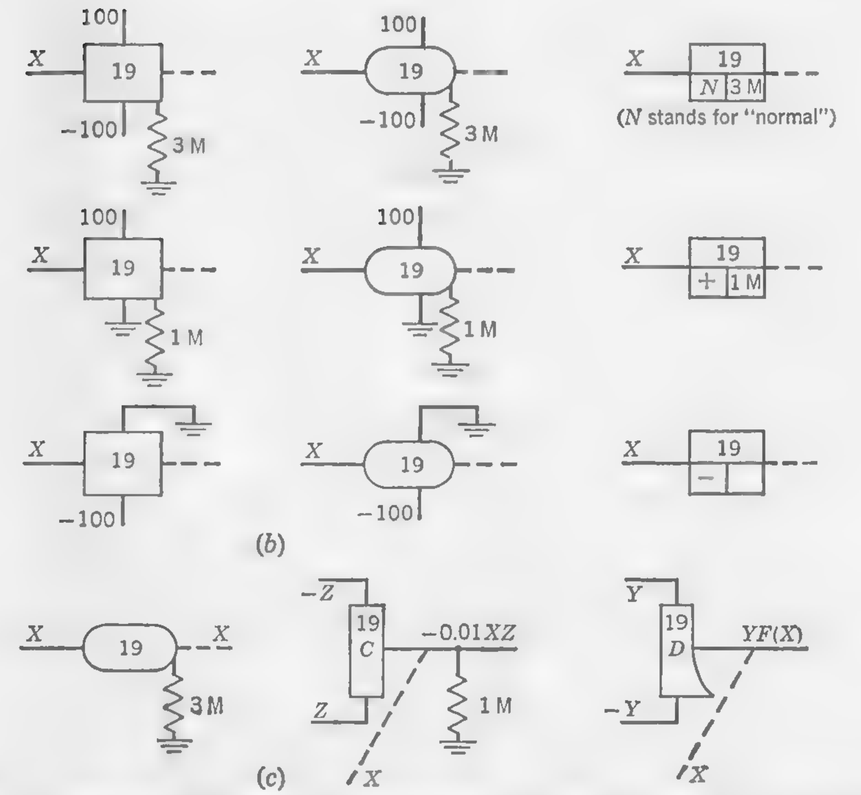


FIG. 1.2.7. Symbols for *computer servomechanisms.* Figure 1.2.7*a* is the proposed IRE standard; the amplifier-motor symbol shown is to replace the simple rectangle used in many earlier drawings. Potentiometer cups are here labeled A, B, C, . . . ; note the load-equalizing resistors on cups A and C (Sec. 3.1). Normally connected ($\pm$100-volt) linear follow-up potentiometers without load-equalizing resistors can be omitted, or one of the notations of Fig. 1.2.7*b* may be used. Special potentiometer tap connections (e.g., center taps) are shown if necessary. Figure 1.2.7*c* shows how servo and potentiometer symbols can be separated in large block diagrams. Two multiplying cups are shown; the follow-up cup is omitted.

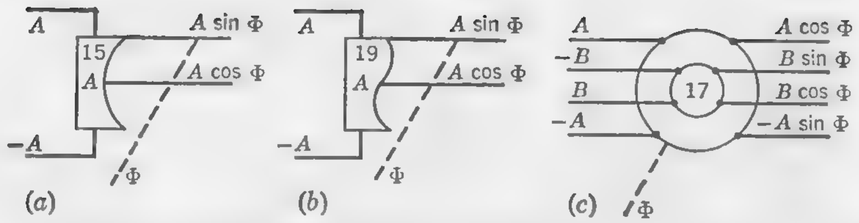


FIG. 1.2.8. Symbols for special *sine-cosine potentiometers* used, for example, in *polar-to-rectangular resolvers*. The resolver servo is represented in the manner of Fig. 1.2.7. Figures 1.2.8*b* and *c* are the proposed IRE standard symbols.

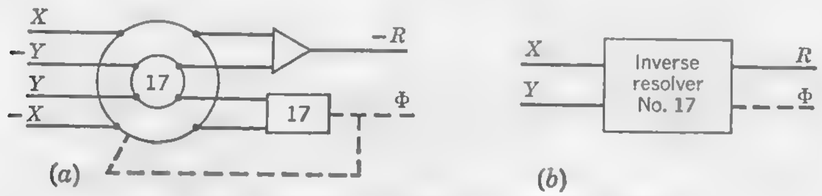


FIG. 1.2.9. Symbols for *rectangular-to-polar resolvers*.

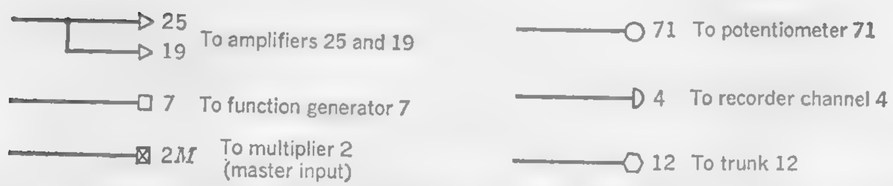


FIG. 1.2.10. Notation showing the destination of signal lines leaving the block diagram. Such a notation may be useful if setup sheets are not used.

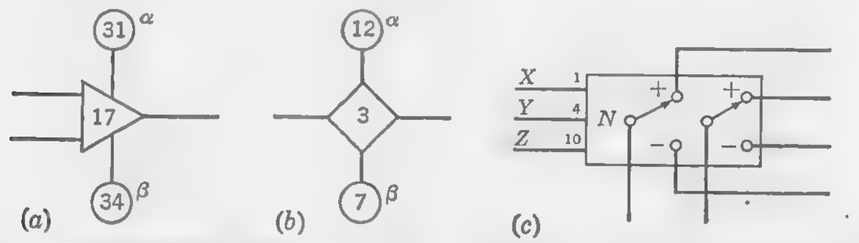


FIG. 1.2.11. These special symbols for feedback-diode-limited amplifiers (*a*) and diode-bridge limiters (*b*) with limits $\alpha > \beta$ are used in some laboratories. Figure 1.2.10*c* shows a suggested symbol for a relay-driving comparator; the relays are in the "+" position if $X + 4Y + 10Z$ is *positive*. In general, "fabricated" diode circuits will have to be shown explicitly.

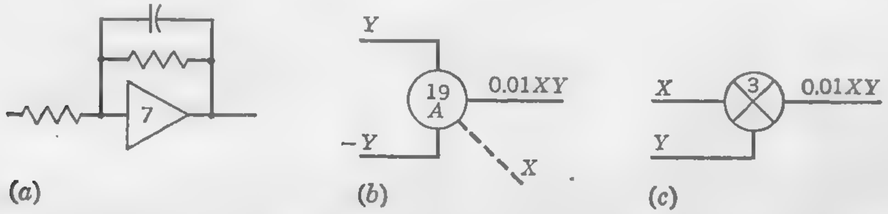


FIG. 1.2.12. These symbols are easily mistaken for other symbols and are not recommended: old high-gain amplifier symbol (*a*), servo-driven potentiometer (*b*), and electronic multiplier (*c*).

Section 2

ELECTRONIC ANALOG COMPUTERS: COEFFICIENT POTENTIOMETERS, OPERATIONAL AMPLIFIERS, AND NETWORKS

By

BERNARD D. LOVEMAN, *Project Engineer, Project CYCLONE, Reeves Instrument Corporation, Garden City, N.Y.*

GRANINO A. KORN, *Professor of Electrical Engineering, University of Arizona, Tucson, Ariz., and* **THERESA M. KORN.**

EDWARD M. BILLINGHURST, *Project Engineer, Beckman/Berkeley Division, Richmond, Calif.*

CHARLES H. SINGLE, *Manager, Computer Engineering, Beckman/Berkeley Division, Richmond, Calif.*

CONTENTS

PART 1: COEFFICIENT-SETTING POTENTIOMETERS

Bernard D. Loveman

Construction and Accuracy

2.1.1. Construction . . . 2–3
2.1.2. Precision vs. Accuracy . . . 2–3
2.1.3. Resolution . . . 2–3
2.1.4. Linearity . . . 2–4
2.1.5. End Resistance . . . 2–4
2.1.6. Frequency-response Characteristics . . . 2–4
2.1.7. Potentiometer Loading . . . 2–4
2.1.8. Resistance Value . . . 2–5

Hand-set Coefficient Potentiometers

2.1.9. Dials . . . 2–5
2.1.10. Comparison-potentiometer Method . . . 2–5
2.1.11. Voltmeter Method . . . 2–6

Servo-set Coefficient Potentiometers

2.1.12. Advantages . . . 2–6
2.1.13. Packaging . . . 2–6
2.1.14. Servo Performance . . . 2–7
2.1.15. Special Features . . . 2–7

Three-terminal Potentiometers

2.1.16. Special Circuits . . . 2–8

PART 2: *RC* NETWORKS AND LINEAR OPERATIONAL AMPLIFIERS

Granino A. Korn and Theresa M. Korn

***RC* Networks and Simple Operational Amplifiers**

2.2.1. Basic Circuits and Design Formulas . . . 2–8
2.2.2. More Detailed Performance Analysis . . . 2–8
2.2.3. Effect of the Load. Output and Input Impedances . . . 2–17
2.2.4. Effects of Gain Changes and Distortion . . . 2–17

More General Operational-amplifier Circuits

2.2.5. Combinations of Operational Amplifiers and Networks..... 2–17
2.2.6. Operational Amplifiers with Four-terminal Input and Feedback Networks..... 2–20
2.2.7. Transfer-function Synthesis by Differential-analyzer Techniques..... 2–20

Errors Due to Parasitic Circuit Parameters

2.2.8. Errors Due to Parasitic Circuit Parameters..... 2–20

PART 3: DESIGN OF VACUUM-TUBE D-C AMPLIFIERS

Edward M. Billinghurst

Design Specifications

2.3.1. Typical Requirements..... 2 40
2.3.2. Operational-amplifier Performance vs. Open-loop Gain and Phase Shift..... 2–41

Design of D-C Amplifier Stages

2.3.3. Choice of Output-stage Circuit..... 2–43
2.3.4. Voltage-amplifier Stages..... 2–43
2.3.5. Selection of Operating Points 2–43
2.3.6. Effects of Cathode-bias Resistors and Screen Series Resistors..... 2 45
2.3.7. The Effect of Load Resistance 2 46
2.3.8. Use of Resistance-coupled-amplifier Tables..... 2–47
2.3.9. Interstage Coupling Networks 2–48

Frequency Stabilization: Roll-off Networks

2.3.10. High-frequency Stabilization. 2–49
2.3.11. Roll-off or Lag-lead Networks 2–49
2.3.12. External Compensating Networks..... 2–51

Drift Compensation and Automatic Balancing Circuits

2.3.13. Offset Voltages and Drift.... 2–51
2.3.14. Chopper Stabilization..... 2–51
2.3.15. Design of Chopper-stabilizer Circuits..... 2–53
2.3.16. Commutator Stabilization.... 2–55

Miscellaneous Design Considerations

2.3.17. Low Grid-current Input Stages 2–56

PART 4: LINEAR-ELEMENT ERROR SOURCES AND EFFECTS

Charles H. Single

Introduction

2.4.1. Scope..... 2–57
2.4.2. Basic Considerations..... 2–57
2.4.3. Dynamic Error vs. Stability Considerations..... 2-57

Computing Resistors

2.4.4. Resistor Specifications and Matching..... 2–58
2.4.5. A-C Characteristics of Wire-wound Resistors..... 2–59

Integrating Capacitors

2.4.6. Properties of Practical Capacitors. Stability..... 2–60
2.4.7. D-C Leakage..... 2–60
2.4.8. Dielectric Losses. Choice of Dielectric Material..... 2–60
2.4.9. Capacitor Equivalent Circuits 2–61
2.4.10. Frequency Dependence of Capacitance..... 2–62
2.4.11. Transient Effects of Dielectric Absorption..... 2–63

Coefficient-setting Potentiometers

2.4.12. Equivalent Circuit for Copper-mandrel Potentiometers..... 2–64
2.4.13. Capacitive Compensation of Copper-mandrel Potentiometers..... 2–66

Combined Optimization of Linear Computing Elements

2.4.14. Capacitive Compensation of Summing Amplifiers..... 2–67
2.4.15. System Performance for a Sine-generating Loop..... 2–70
2.4.16. Summary..... 2–72

PART 5: EXAMPLES OF COMPUTER AMPLIFIER AND POWER-SUPPLY CIRCUITS

Granino A. Korn

2.5.1. Examples of Computer Amplifier Circuits..... 2–72
2.5.2. High-voltage Power Supplies.. 2–82
2.5.3. Reference Power Supplies..... 2–82

PART 1: COEFFICIENT-SETTING POTENTIOMETERS

By Bernard D. Loveman

CONSTRUCTION AND ACCURACY

2.1.1. Construction. *Coefficient potentiometers* (*scale-factor potentiometers, attenuators*) multiply machine voltages by constant coefficients between 0 and 1 (Fig. 1.2.1). Each potentiometer consists of a resistance element with terminals at each end (usually denoted as HI and LO) and a sliding contact or brush (ARM terminal) which moves along the resistance element. A desired fraction of the input voltage applied across the HI and LO terminals appears as the output voltage between the ARM and LO terminals; the latter is grounded in most applications.

The resistance element may consist of (1) a film of carbon or other conducting material deposited on an insulated form, or (2) a number of turns of resistance wire wound on a core (mandrel). Single-turn potentiometers may have a vernier adjustment in the form of a low variable resistance in series with the potentiometer. Most commercial d-c analog computers utilize wire-wound potentiometers.

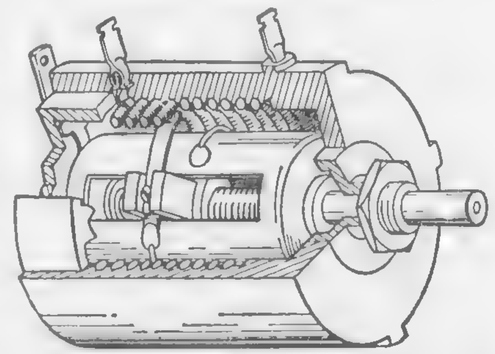


Fig. 2.1.1. Construction of a 10-turn helical-coil wire-wound potentiometer. (*Gibbs Micropot; from Chance et al., Waveforms, McGraw-Hill Book Company, Inc., New York, 1949.*)

The resistance element, and hence the brush motion, may be linear, circular, or helical. Linear-motion potentiometers take very little panel space (small cross section), but high accuracy would require considerable depth behind the panel. Circular- and helical-mandrel potentiometers offer a reasonable compromise between panel space and depth. Their maximum diameter is usually limited to 2 in.; if desired, two or more potentiometers can be ganged. Most d-c analog computers use 10-turn helical-element wire-wound potentiometers with resistance values ranging between 10,000 and 100,000 ohms (Fig. 2.1.1). The following discussion refers specifically to potentiometers of this general type; Table 2.1.1 shows typical specifications.

Table 2.1.1. Specifications of a Typical Coefficient Potentiometer

No. of turns	10
Resolution	0.009%
Linearity	0.05%
Power rating at 40°C	5 watts
Resistance	30 kilohms ± 5%
Mechanical rotation	3,600 (+4 deg/−0 deg)
Electrical rotation	3,600 (+4 deg/−0 deg)
End resistance (HI end)	3 ohms (max)
End resistance (LO end)	30 ohms (max)

2.1.2. Precision vs. Accuracy. *Precision* refers to the number of places to which a coefficient can be set or read repeatedly. *Accuracy* is a measure of the conformity of the coefficient to a specified ratio, i.e., the number of places which correspond to a given standard. The coefficient accuracy can be no greater than the precision. Precision and accuracy of coefficient potentiometers are always specified in per cent of the full potentiometer scale.

2.1.3. Resolution. The precision to which a coefficient can be set depends on the *resolution* of the potentiometer. The number of wire turns N determines the resolu-

tion; the latter is usually expressed in per cent as $100/N$. The resolution sets the upper limit for the precision to which the coefficient can be set.

2.1.4. Linearity. In some computers, one sets coefficients by reading the shaft position on a linear dial, so that the output voltage must be a linear function of shaft position. The amount that the output deviates from this straight line is called the *linearity accuracy*. Since the linearity is not as good as the resolution, the coefficient accuracy is limited by the *potentiometer linearity*, which is discussed in detail in Art. 3.1.2.

2.1.5. End Resistance. If the potentiometer shaft is set at the LO (or HI) end, there will be a resistance between the ARM and the LO (or HI) terminal. These end resistances determine the minimum (or maximum) coefficient obtainable. The end resistance of a computer potentiometer may be as large as 0.5 per cent of the total resistance. Smaller end resistances can usually be specified at increased cost. One possible compromise is to have small end resistance at the LO end and standard end resistance at the HI end.

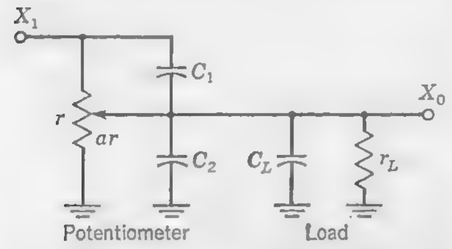


FIG. 2.1.2. A simplified equivalent circuit for a multiturn copper-mandrel potentiometer (Beckman Helipot®). The transfer function of this circuit replaces that of Eq. (2.1.1); the circuit is useful for potentiometer settings a between 0.05 and 0.95 and frequencies less than $1/24\pi rC_1$, with r between 10 and 100 kilohms. For type A Helipots, $C_1 = C_2$ and varies between 200 $\mu\mu$f (r = 10 kilohms) and 225 $\mu\mu$f (r = 100 kilohms). C_L is the load capacitance and includes wiring, etc. (*From C. H. Single, Low-frequency Equivalent Circuit for Copper-mandrel Potentiometers, Report CRD 58-14, Berkeley Division, Beckman Instruments, Inc., Richmond, Calif., Sept. 3, 1958.*)

2.1.6. Frequency-response Characteristics. In addition to resolution and linearity the dynamic characteristics of the potentiometer and its cabinet wiring limit the accuracy attainable. The construction of the potentiometer determines its frequency characteristics; the factors that enter are:

1. Inductance of the wire turns
2. Capacitance between individual turns of wire
3. Capacitance between the resistance wire and the mandrel

The deterioration in frequency performance is due primarily to the third factor and depends on potentiometer resistance. For resistances between 10,000 and 100,000 ohms the distributed mandrel-to-winding capacitance varies from 0.002 to 0.004 μf. The bandwidth of a potentiometer is defined as the frequency at which the maximum in-phase error is 0.1 per cent of full scale. For potentiometer resistances from 10,000 to 100,000 ohms the bandwidth varies from 15 to 1 kc.

Furthermore, capacitance is also added if shielded wire is used when the potentiometer is installed in the computer. The over-all effect is to change the attenuation factor slightly; but, more important, a quadrature voltage is introduced, i.e., the input and output voltage are out of phase. This phase shift should be kept as small as possible. Even a phase shift of only 0.01 deg at computing frequencies may produce errors in sensitive computer circuits such as a sine-generating loop (Fig. 2.1.2; see also Sec. 2.4).

2.1.7. Potentiometer Loading. Figure 2.1.3 shows the most commonly used coefficient-potentiometer circuit. The coefficient depends not only on the shaft position of the potentiometer but also upon the load resistance r_L; r_L is usually "grounded" at an amplifier summing junction. This loading effect must be taken into account when the potentiometer is set. Two techniques are used, depending upon the setting method. The first consists of calculating (or using a graph, Fig. 2.1.4) the necessary correction. In the second method the input is disconnected and a reference voltage (+100 volts) is substituted; then the potentiometer shaft is turned until the desired output is obtained, with the load connected (Arts. 2.1.10 and 2.1.11). To prevent the output amplifier from overloading (which would, incidentally, produce an erroneous

loading effect) most computers provide a control mode (POT SET) in which r_L is disconnected from the amplifier and grounded (Fig. 2.1.3).

2.1.8. Resistance Value. Theoretically, the resistance of a coefficient potentiometer is not significant since it is used as a voltage divider. Practically, however, the range of resistance values is restricted. If the resistance is too low, it will load the input amplifier excessively and will not have adequate resolution. The upper limit is determined by the availability of wire-wound resistance elements and by the phase shift due to capacitance from the arm to ground. The resistance of potentiometers used in commercial computers varies from 10,000 to 100,000 ohms.

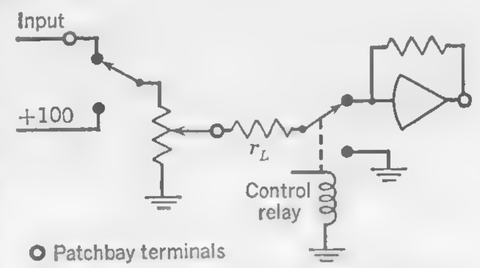


FIG. 2.1.3. Circuit for setting a potentiometer with the load connected.

The lower limit on the power rating is computed from the requirement that the potentiometer must take the reference voltage (+100 volts). Four- or five-watt units provide a satisfactory margin of safety.

Failures in coefficient potentiometers usually arise not from excessive input voltage but rather from incorrect patching of the output (operator mistake). For example, a 10,000-ohm 5-watt potentiometer has a current rating of 13 ma. If the input is 100 volts, the coefficient is 0.95, and the output (arm) is grounded, then 60 ma flows through some of the wire turns. Since mistakes of this kind arise, it is advisable to fuse the potentiometer arm. This protection is required only for accidental overloads, and thus a fuse capacity of two or three times rated current is adequate. A larger fuse also reduces the loading effect of the fuse resistance. For the example cited above a $\frac{1}{32}$-amp fuse is provided.

FIG. 2.1.4. Potentiometer-loading correction chart based on the relation

$$a - \frac{X_0}{X_1} \approx a^2(1 - a)\,\frac{r}{r_L}$$

Multiply the ordinates corresponding to a desired value of $x = X_0/X_1$ by the value of $r/r_L \leq 0.3$; this quantity must be added to the desired value of the coefficient X_0/X_1 to obtain the correct potentiometer setting a. The use of charts of this type is not recommended if setting accuracies better than 0.1 per cent are desired; in this case charts or tables based on the exact formula (2.1.1) should be used. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York., 1956.*)

HAND-SET COEFFICIENT POTENTIOMETERS

2.1.9. Dials. One method for hand-setting coefficient potentiometers is to provide each with a calibrated linear dial and a shaft lock. Turn-counting analog dials (Helipot DUODIALS®) and dials with in-line digital indication (Borg Model 1301 MICRODIALS®) are available for use with multiturn potentiometers.

The dial reading a corresponding to a given coefficient X_0/X_1 depends on the ratio of the potentiometer resistance r to the load resistance r_L. The *loading correction* (which must be added to the desired coefficient to obtain the correct dial setting) is given by

$$a - \frac{X_0}{X_1} = \frac{a^2(1 - a)(r/r_L)}{1 + a(1 - a)(r/r_L)} \qquad (2.1.1)$$

Figure 2.1.4 shows a universal loading-correction curve based on Eq. (1); one may also tabulate the loading corrections for suitable values of the ratio r/r_L.

2.1.10. Comparison-potentiometer Method. Figure 2.1.5 shows a circuit for setting coefficient potentiometers, with their loads connected, by a null-indication

comparison with a standard potentiometer set to the desired coefficient value. The null detector used may be a galvanometer, vacuum-tube voltmeter, or oscilloscope and should have a sensitivity of 10 mv or better. The null detector must not load either the coefficient potentiometer or the comparison potentiometer. The comparison circuit is also useful for accurate measurements of other computer voltages.

2.1.11. Voltmeter Method. A more convenient if somewhat more expensive method replaces the comparison circuit in Fig. 2.1.5 by a precision digital or servo voltmeter which reads each coefficient setting directly, with the load connected. The voltmeter input impedance must be sufficiently high to prevent loading errors.

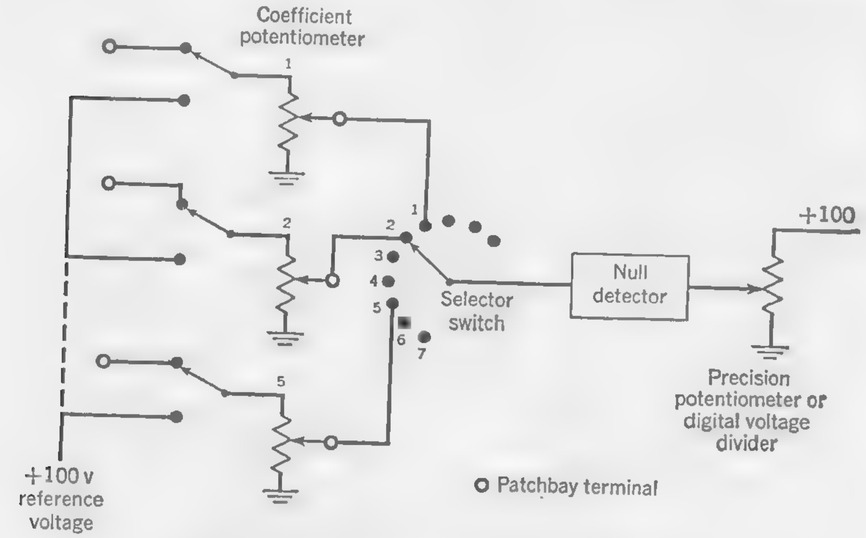


FIG. 2.1.5. Comparison-potentiometer circuit.

SERVO-SET COEFFICIENT POTENTIOMETERS

2.1.12. Advantages. In large computers (over 100 amplifiers) the use of servo-set potentiometers becomes economical. Servo-set systems are inherently similar to the comparison system described above, except that each potentiometer shaft is turned by a servomechanism controlled from a central keyboard. Such systems are *not* designed merely to prevent operator fatigue; the effort of setting one hundred potentiometers by hand is not much greater than that required to push a set of buttons one hundred times. On the other hand, servo-set systems permit punched-paper-tape input and furnish a printed output, and are thus powerful aids in setting up large problems. A block diagram of one system is shown in Fig. 2.1.6. Such systems provide the following advantages:

1. Each coefficient setting is automatically and permanently recorded.
2. Every coefficient change is recorded.
3. Each time a problem is removed all coefficients are recorded, and a punched paper tape is prepared so that all coefficients can be correctly restored.

2.1.13. Packaging. The components shown in Fig. 2.1.6 may be packaged in a variety of ways. The input-output equipment can be combined in one unit, an electric typewriter with a paper-tape punch, tape reader, and a translator. The decoder may consist of relays and crossbar, stepping, or electronic switches, as well as combinations of these components.

Each coefficient potentiometer may be driven by its own servo motor, or several potentiometers (usually 20) may be connected to a single motor by means of clutches; new inexpensive clutches have made the second method economical. Two precautions must be observed in systems using clutches. First, the potentiometer should be declutched after servo null is reached but before the output voltage is read. Second, clutches must decouple completely to prevent the motion of one potentiometer from dragging others.

2.1.14. Servo Performance. In addition to the standard servo-design criteria, a few special factors must be considered. The system error is usually specified to be less than 10 or 20 mv. Since component errors can arise in several units, the servo must meet the stringent requirement that its static error be less than 10 mv. Coefficient potentiometers having about 12,000 wire turns must be set to an accuracy of one wire turn.

The servo dynamic response is also critical; the potentiometer should be set rapidly, but at the same time the servo should be overdamped to prevent potentiometer wear

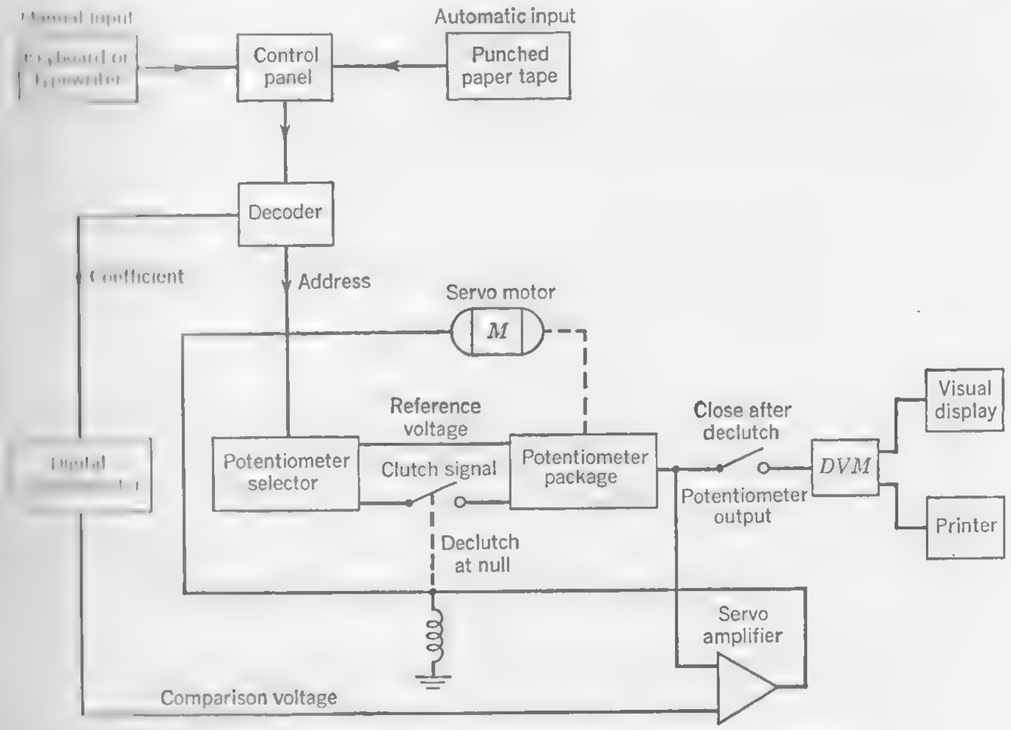


FIG. 2.1.6. Block diagram of a servo-set potentiometer system.

due to hunting. One method for accomplishing both design objectives is to provide nonlinear damping.

2.1.15. Special Features. In addition to the basic circuit described most systems include some special control circuits:

Error circuits to warn the operator if the coefficient measured by the digital voltmeter does not agree with the digital potentiometer output
Circuits to measure potentiometer outputs when the potentiometers are connected in the problem
Provision for the slaving of any servo-set potentiometer to a manual potentiometer
Interlocks to prevent the system from interpreting erroneous codes

As in the other comparison methods, this system contains all the essential components for measuring other voltages in the computer such as amplifier outputs. Selectors, controls, and coding to facilitate these measurements can be added at relatively small additional cost. Furthermore, by the addition of circuits to control the computer (RESET, OPERATE, HOLD, etc.) semiautomatic and automatic operation can be achieved (Sec. 4.2).

THREE-TERMINAL POTENTIOMETERS

2.1.16. Special Circuits. In many d-c analog computers at least some of the coefficient potentiometers have free rather than grounded LO terminals for use in various special feedback circuits and diode circuits (Sec. 3.3). In many such connections, the 100-volt reference voltage used with comparison, voltmeter, and servo-setting methods would blow the potentiometer fuse, so that dial readings or indirect voltage indications are used to set the potentiometer. If servo setting is essential, one can disconnect the potentiometer from its computing connection during the setting procedure.

References

1. Logan, B. F.: AC Performance and Phase Compensation of Copper-mandrel Potentiometers, *D.A.C.L. Mem.* R. M. 7187-7, MIT, Cambridge, Mass.
2. McDonald, R., and I. Hogan: Accuracy of Potentiometer Linearity Measurements, *Tele-Tech & Electronic Inds.*, August, 1953.
3. Schmidt, H. A.: The Precision Potentiometer as a Voltage Divider, *Product Eng. Handbook*, 1954.

PART 2: *RC* NETWORKS AND LINEAR OPERATIONAL AMPLIFIERS

By GRANINO A. KORN *and* THERESA M. KORN

RC NETWORKS AND SIMPLE OPERATIONAL AMPLIFIERS

2.2.1. Basic Circuits and Design Formulas.[1,2] Relatively simple *resistance-capacitance networks* permit the computer designer to relate input and output voltages by the basic mathematical operations of multiplication by constants, addition, integration, and differentiation (Fig. 2.2.1) and by more general linear transfer-function operators of interest in control-system synthesis (Table 2.2.1). Useful particularly in special-purpose computers, computing networks can often save electronic circuits and their attendant power-supply, drift, and maintenance problems. Suitable network components are discussed in Sec. 2.4.

Parallel-feedback operational amplifiers offer much greater design flexibility, freedom from attenuation and loading effects (high input impedance and low output impedance), simpler calibration, and isolation of input and output. Operational amplifiers use *RC* feedback networks together with d-c amplifiers; modern chopper-stabilized d-c amplifiers permit accuracies comparable to those of networks alone.

The operational-amplifier circuits of Fig. 2.2.1 are all special cases of the basic circuit of Fig. 2.2.2. If there is appreciable admittance between summing point and ground, this is regarded as an additional impedance $Z_k = Z_g$ without an input voltage (Sec. 2.3). If the feedback loop is stable, and the loop gain is large, the *summing-point voltage* X_0/A is negligible, so that node analysis yields the basic design formula

$$X_0 = -Z_0(P)\left[\frac{X_1}{Z_1(P)} + \frac{X_2}{Z_2(P)} + \cdots\right] \tag{2.2.1}$$

where $P \equiv d/dt$ denotes differentiation with respect to real time (computer time). Table 2.2.2 shows a number of useful circuits.

2.2.2. More Detailed Performance Analysis (see also Secs. 2.3 and 2.4).[1] If X_0/A is not neglected, one must add an *error term*

$$e(t) = \frac{1}{1 - A(P)\beta(P)}\, Z_0(P)\left[\frac{X_1}{Z_1(P)} + \frac{X_2}{Z_2(P)} + \cdots\right]$$
$$\beta(P) = \left[1 + \frac{Z_0(P)}{Z_1(P)} + \frac{Z_0(P)}{Z_2(P)} + \cdots\right]^{-1} \tag{2.2.2}$$

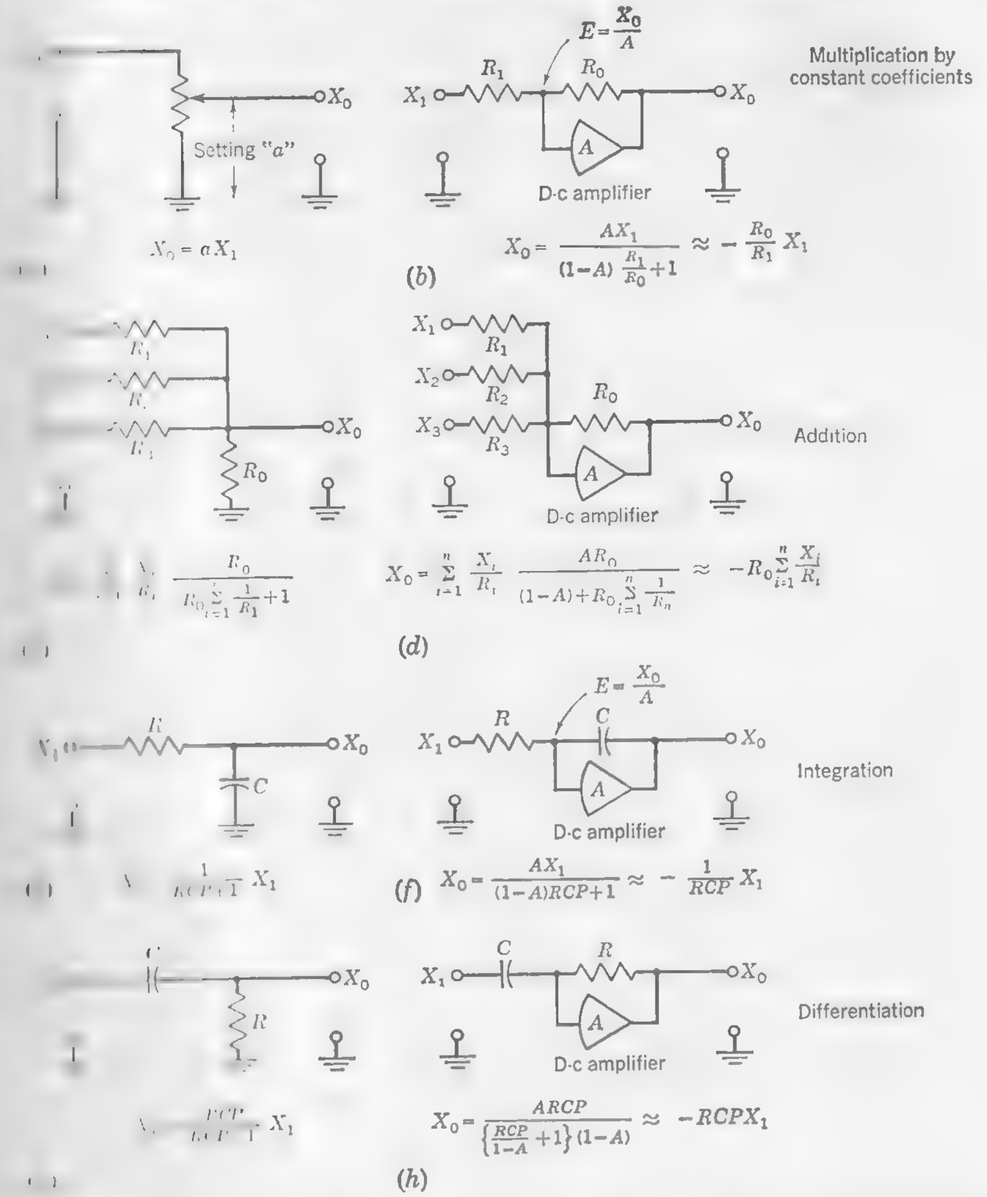
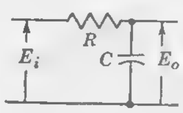
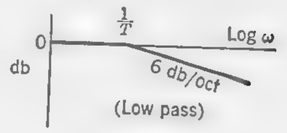
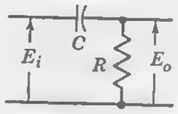
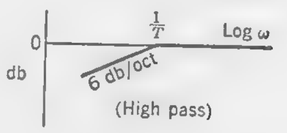
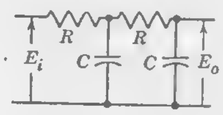
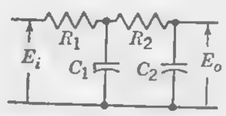
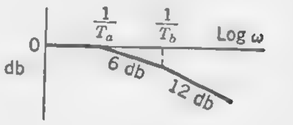
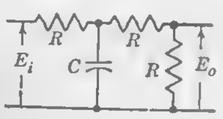
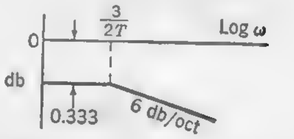
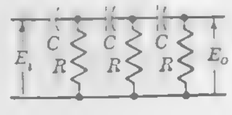
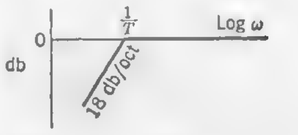
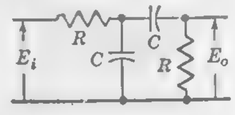
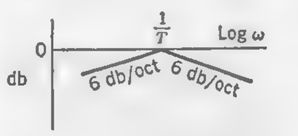
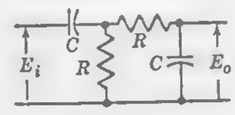
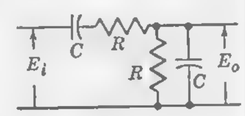
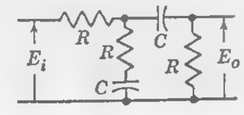
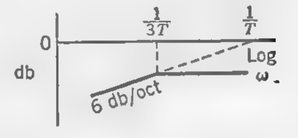


Fig. 2.2.1. Simple computing networks and operational amplifiers. In the differentiating amplifier of Fig. 2.2.1*h*, a small "smoothing" capacitor across R will reduce output noise, and a small resistor in series with C will prevent overload of a preceding amplifier; the resulting transfer function is that of circuit 7 in Table 2.2.2. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

[illegible] expression (1) for the output voltage X_0. The operator $A(P)$ is the effective [illegible] forward gain (allowing for amplifier loading). For sinusoidal input, substitution of $P = j\omega$ in Eq. (2) yields the *per cent error amplitude* (per cent of *correct* value) and the *phase error* involved in the simplified performance equation (1):

$$|\epsilon_\omega| = \frac{100}{|1 - A(j\omega)\beta(j\omega)|} \qquad \delta_\omega = -\arg\left[\frac{1}{A(j\omega)\beta(j\omega)} - 1\right] \tag{2.2.3}$$

Table 2.2.1. Typical *RC* Networks*

Network	Transfer function	Gain curve
(1) 	$\frac{1}{1+Ts}$ $\phi = -\arctan T\omega$	
(2) 	$\frac{Ts}{1+Ts}$ $\phi = 90° - \arctan T\omega$	
(3) 	$\frac{1}{1+3Ts+T^2s^2}$ $\phi = -\arctan \frac{3T\omega}{1-T^2\omega^2}$	
(4) 	$\frac{1}{1+(T_1+T_2+R_1C_2)s+T_1T_2s^2}$, $T_1 = R_1C_1$, $T_2 = R_2C_2$ (Roots of denominator quadratic are T_A, T_B) $\phi = -\arctan \frac{(T_1+T_2+R_1C_2)\omega}{1-T_1T_2\omega^2}$	
(5) 	$\frac{1}{3+2Ts}$ $\phi = -\arctan \frac{2T\omega}{3}$	
(6)	[illegible]	
(7) 	$\frac{T^3s^3}{1+5Ts+6T^2s^2+T^3s^3}$ $\phi = 270° - \arctan \frac{(5-T^2\omega^2)T\omega}{1-6T^2\omega^2}$	
(8) 	$\frac{Ts}{1+3Ts+T^2s^2}$ $\phi = 90° - \arctan \frac{3T\omega}{1-T^2\omega^2}$	
(9) 	$\frac{Ts}{1+3Ts+T^2s^2}$ $\phi = 90° - \arctan \frac{3T\omega}{1-T^2\omega^2}$	
(10) 	$\frac{Ts}{1+3Ts+T^2s^2}$ $\phi = 90° - \arctan \frac{3T\omega}{1-T^2\omega^2}$	
(11) 	$\frac{Ts}{1+3Ts}$ $\phi = 90° - \arctan 3T\omega$	

* Extracted from E. W. Tschudi, Transfer Functions for R-C and R-L Equalizers, *Electronics Buyers' Guide*, June, 1953, p. R-61.

Table 2.2.1. Typical *RC* Networks (*Continued*)

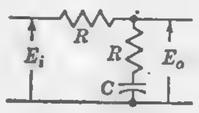
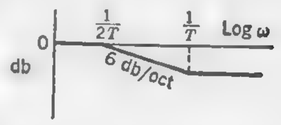
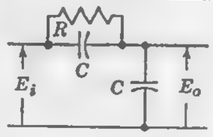
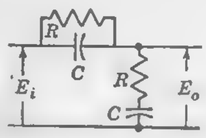
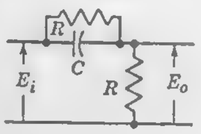
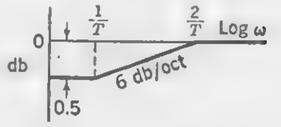
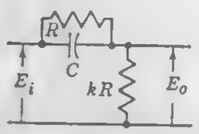
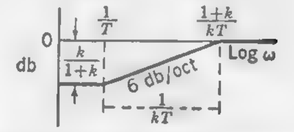
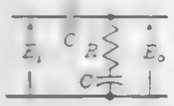
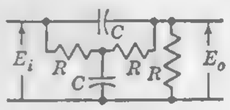
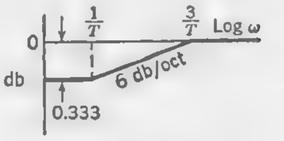
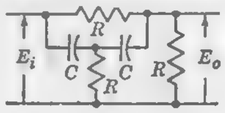
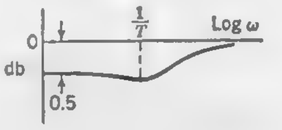
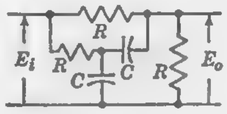
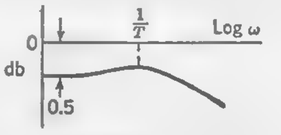
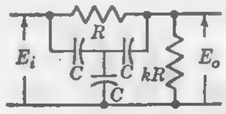
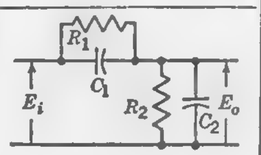
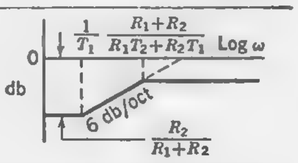
Network	Transfer function	Gain curve
(12) 	$\frac{1+Ts}{1+2Ts}$ $\phi = \arctan T\omega - \arctan 2T\omega$	
(13) 	$\frac{1+Ts}{1+2Ts}$ $\phi = \arctan T\omega - \arctan 2T\omega$	
(14) 	$\frac{(1+Ts)^2}{1+3Ts+T^2s^2}$ $\phi = 2\arctan T\omega - \arctan \frac{3T\omega}{1-T^2\omega^2}$	
(15) 	$\frac{1+Ts}{2+Ts}$ $\phi = \arctan T\omega - \arctan \frac{T\omega}{2}$	
(16) 	$\frac{k(1+Ts)}{(1+k)+kTs}$ $\phi = \arctan T\omega - \arctan \frac{kT\omega}{1+k}$	
	$\frac{1-Ts}{2+Ts}$ $\phi = \arctan T\omega - \arctan \frac{T\omega}{2}$	
(18) 	$\frac{1+Ts}{3+Ts}$ $\phi = \arctan T\omega - \arctan \frac{T\omega}{3}$	
(19) 	$\frac{(1+Ts)^2}{2+5Ts+T^2s^2}$ $\phi = 2\arctan T\omega - \arctan \frac{5T\omega}{2-T^2\omega^2}$	
(20) 	$\frac{1+3Ts}{2+5Ts+T^2s^2}$ $\phi = \arctan 3T\omega - \arctan \frac{5T\omega}{2-T^2\omega^2}$	
(21) 	$\frac{k(3+Ts)}{3(1+k)+2kTs} = \frac{k[9(1+k)+2k\omega^2T^2+3(1-k)j\omega T]}{9(1+k)^2+4k^2\omega^2T^2}$ $\phi = \arctan \frac{T\omega}{3} - \arctan \frac{2kT\omega}{3(1+k)}$	j Term negative for $k > 1$, zero for $k = 1$, positive for $k < 1$ $A = k\sqrt{\frac{9+T^2\omega^2}{9(1+k)^2+4k^2T^2\omega^2}}$
(22) 	$\frac{R_2(1+T_1s)}{(R_1+R_2)+(R_1T_2+R_2T_1)s}$ $\phi = \arctan T_1\omega - \arctan \frac{(R_1T_2+R_2T_1)\omega}{R_1+R_2}$	

Table 2.2.1. Typical *RC* Networks *(Continued)*

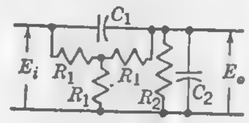
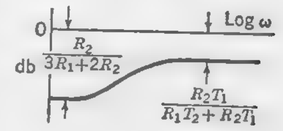
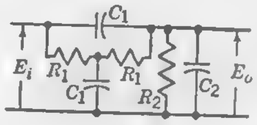
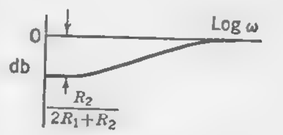
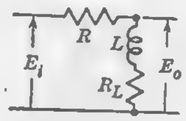
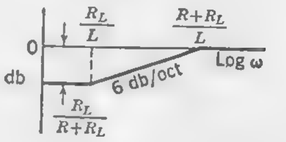
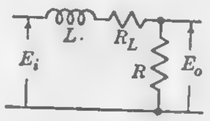
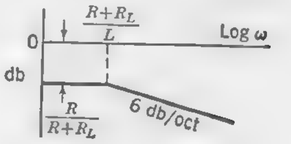
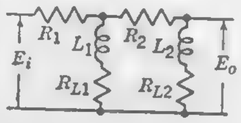
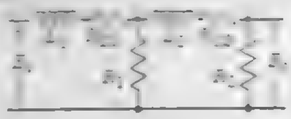
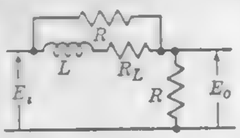
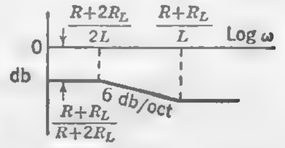
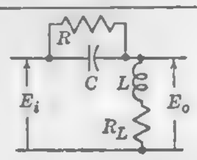
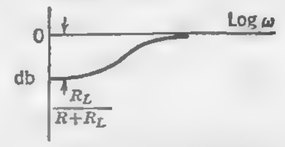
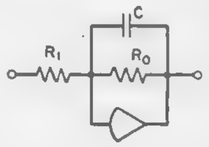
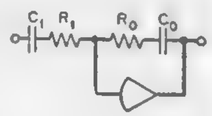
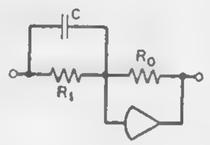
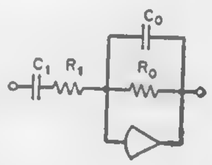
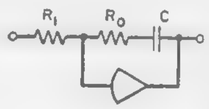
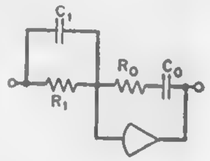
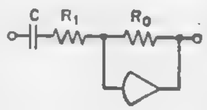
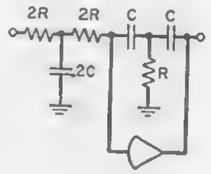
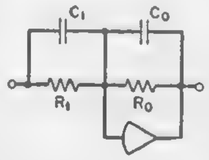
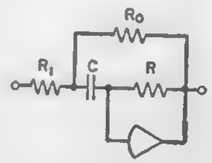
Network	Transfer function	Gain curve
(23) 	$\dfrac{R_2(1+3T_1s)(1-T_2s)}{(3R_1+2R_2)+R_2(3T_1-2T_2)s+3T_2(R_1T_2+R_2T_1)s^2}$ $\phi = \arctan 3T_1\omega - \arctan T_2\omega - \arctan \dfrac{R_2(3T_1-2T_2)\omega}{3R_1+2R_2-3T_2(R_1T_2+R_2T_1)\omega^2}$	
(24) 	$\dfrac{R_2[1+(2T_1-T_2)s+T_1(T_1-2T_2)s^2-T_1^2T_2s^3]}{(2R_1+R_2)+(R_1T_1+3R_2T_1-R_2T_2)s+(R_2T_1^2-3R_2T_1T_2-2R_1T_2^2)s^2+T_1(R_1T_2^2+R_2T_1T_2)s^3}$ $\phi = \arctan \dfrac{(2T_1-T_2+T_1^2T_2\omega^2)\omega}{1-T_1(T_1-2T_2)\omega^2} - \arctan \dfrac{[R_1T_1+3R_2T_1-R_2T_2-T_1(R_1T_2^2+R_2T_1T_2)\omega^2]\omega}{2R_1+R_2-(R_2T_1^2-3R_2T_1T_2-2R_1T_2^2)\omega^2}$ (Leading angle)	
(25) 	$\dfrac{R_L+Ls}{R+R_L+Ls}$ $\phi = \arctan \dfrac{L\omega}{R_L} - \arctan \dfrac{L\omega}{R+R_L}$	
(26) 	$\dfrac{R}{R+R_L+Ls}$ $\phi = \arctan \dfrac{L\omega}{R+R_L}$	
(27) 	$\dfrac{(R_{L1}+L_1s)(R_{L2}+L_2s)}{[(R_1+R_{L1})(R_2+R_{L2})+R_1L_1]+[(R_1+R_{L1})L_2+(R_2+R_{L2})L_1+R_1L_1]s+L_1L_2s^2}$ $\phi = \arctan \dfrac{L_1\omega}{R_L} + \arctan \dfrac{L_2\omega}{R_{L2}} - \arctan \dfrac{[(R_1+R_{L1})L_2+(R_2+R_{L2})L_1+R_1L_1]\omega}{(R_1+R_{L1})(R_2+R_{L2})+R_1L_1-L_1L_2\omega^2}$	
	[illegible]	
	$\dfrac{R+R_L+Ls}{R+2R_L+2Ls}$ $\phi = \arctan \dfrac{L\omega}{R+R_L} - \arctan \dfrac{2L\omega}{R+2R_L}$	
(30) 	$\dfrac{(R_L+Ls)(1+Ts)}{(R+R_L)+(R_LT+L)s+LTs^2}$ $\phi = \arctan \dfrac{L\omega}{R_L} + \arctan T\omega - \arctan \dfrac{(R_LT+L)\omega}{R+R_L-LT\omega^2}$	

Table 2.2.2. Examples of Parallel-feedback-type Operational Amplifiers for a Number of Transfer Functions $\frac{X_o}{X_1} = -\frac{Z_o(P)}{Z_1(P)}$

Additional circuits may be designed with the aid of Table 2.2.4.

	Transfer function X_o/X_1	Circuit		Transfer function X_o/X_1	Circuit
1	$-\frac{R_o}{R_1}\frac{1}{R_oCP+1}$		6	$-\frac{C_1}{C_o}\frac{R_oC_oP+1}{R_1C_1P+1}$	
2	$-\frac{R_o}{R_1}(R_1CP+1)$		7	$-\frac{R_oC_1P}{(R_oC_oP+1)(R_1C_1P+1)}$	
3	$-\frac{R_oCP+1}{R_1CP}$		8	$-\frac{(R_oC_oP+1)}{(R_1C_1P+1)}\frac{}{R_1C_oP}$	
4	$-\frac{R_oCP}{R_1CP+1}$		9	$-\frac{1}{4(RC)^2P^2}$	
5	$-\frac{R_o}{R_1}\frac{R_1C_1P+1}{R_oC_oP+1}$		10	$-\frac{R_oRCP}{(R_o+R)R_1CP+R_1+R_o}$	

These relations are used to estimate the working frequency range of the operational amplifier (*see also* Sec. 2.3; note that *error amplitude* is not the same as *amplitude error*).

2.2.3. Effect of the Load. Output and Input Impedances.[1] If a load of impedance Z_L is connected to the operational amplifier, the effective d-c amplifier forward gain is

$$A(j\omega) = \frac{A_0(j\omega) + (Z_I/Z_0)}{1 + (Z_I/Z_0) + (Z_I/Z_L)} \tag{2.2.4}$$

where $A_0(j\omega)$ and Z_I are, respectively, the open-circuit gain and the internal impedance (output impedance, source impedance) of the d-c amplifier alone.

The *internal impedance* (*output impedance, Thévenin source impedance*) Z_s of the operational amplifier is

$$Z_s = \frac{Z_0 Z_I}{[1 - \beta(j\omega)]Z_I + [1 - A_0(j\omega)\beta(j\omega)]Z_0}$$

$$\left[\approx -\frac{Z_I}{A_0(j\omega)\beta(j\omega)} \quad \text{if } |A_0(j\omega)\beta(j\omega)| \gg 1\right] \tag{2.2.5}$$

In most applications, Z_I varies between 1,000 and 50,000 ohms, and $|A_0(j\omega)\beta(j\omega)|$ is so large that $|Z_s/Z_L|$, and thus the effect of the load, is negligible at low frequencies. At higher frequencies, capacitive loading may have to be considered.

The *input impedance* seen by the ith input voltage source is practically equal to the impedance Z_i whenever the absolute value of the loop gain $A(j\omega)\beta(j\omega)$ is large.

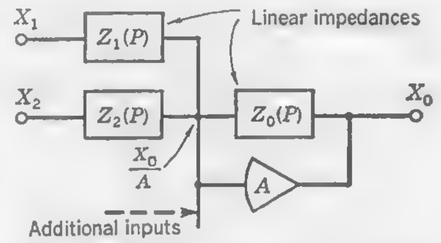


FIG. 2.2.2. Basic parallel-feedback operational amplifier. $Z_0(P)$, $Z_1(P)$, $Z_2(P)$, . . . are two-terminal impedance operators, with $P = d/d\tau$. The d-c forward gain $A(0)$ is negative for stability.

2.2.4. Effects of Gain Changes and Distortion. The inverse feedback inherent in the operational-amplifier circuit helps to reduce the effects of gain changes in the d-c amplifier on the output voltage X_0; the range of permissible gain changes may be investigated with the aid of Eqs. (2) and (3).

For pure sinusoidal voltages, the fractional effect of a steady-state gain change $dA(j\omega)$ on the output amplitude is reduced in the ratio $1:|1 - A(j\omega)\beta(j\omega)|$. A similar reduction applies to the fractional amplitude of each sinusoidal distortion and intermodulation component originating in the d-c amplifier; the effect of the feedback on *total* distortion is not easily computed.

Effects of *d-c amplifier drift* and *leakage* are discussed in Sec. 2.3.

MORE GENERAL OPERATIONAL-AMPLIFIER CIRCUITS

2.2.5. Combinations of Operational Amplifiers and Networks.[1,2] Table 2.2.3 shows a number of computer circuits capable of implementing complex dynamic-system relations with a single d-c amplifier. The special circuit of Fig. 2.2.3 is particularly useful for economical computer representation of second-order linear systems (mass-spring-dashpot systems) with the transfer function

$$\frac{X_0}{X_1} = -\frac{1}{\frac{s^2}{\omega_n^2} + 2\frac{\xi}{\omega_n}s + 1} = -\frac{1}{R_1R_2C_1C_2s^2 + (R_1 + 2R_2)C_2s + 1} \tag{2.2.6}$$

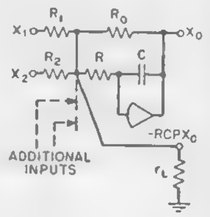
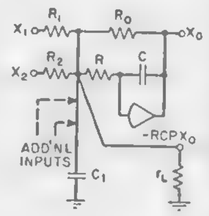
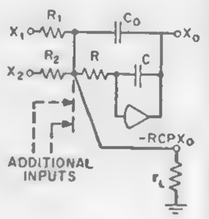
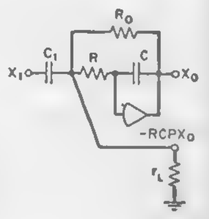
with the *damping ratio* ξ and the *undamped natural frequency* $\omega_n/2\pi$ given by

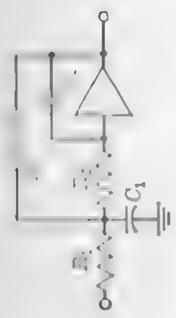
$$\xi = \sqrt{\frac{(R_1 + 2R_2)^2}{4R_1R_2}\frac{C_2}{C_1}} \geq \sqrt{\frac{2C_2}{C_1}} \qquad \omega_n = \sqrt{\frac{1}{R_1R_2C_1C_2}} \tag{2.2.7}$$

Reference 2 contains several useful examples.

Table 2.2.3. Additional Operational-amplifier Circuits

These circuits may simulate components of dynamical or electrical systems and are useful for the solution of partial differential equations. Each circuit involves a simple integration and yields $-RCPX_o$ as well as X_o. The load resistance r_L is usually an input resistor of another operational amplifier. These circuits are most useful if frequent parameter changes are not required.

	Machine equation	Circuit
2.1	$X_o = -\left[\dfrac{R_o}{\left(\frac{1}{R}+\frac{1}{r_L}+\frac{1}{R_o}+\frac{1}{R_1}+\frac{1}{R_2}+\cdots\right)R_oRCP+1}\right] \times\left(\frac{X_1}{R_1}+\frac{X_2}{R_2}+\cdots\right)$	
2.2	$X_o = -\left[\dfrac{R_o}{R_oC_1RCP^2+\left(\frac{1}{R}+\frac{1}{r_L}+\frac{1}{R_o}+\frac{1}{R_1}+\frac{1}{R_2}+\cdots\right)R_oRCP+1}\right] \times\left(\frac{X_1}{R_1}+\frac{X_2}{R_2}+\cdots\right)$	
2.3	$X_o = -\left[\dfrac{1}{RCP\left(C_oP+\frac{C_o}{RC}+\frac{1}{R}+\frac{1}{r_L}+\frac{1}{R_1}+\frac{1}{R_2}+\cdots\right)}\right] \times\left(\frac{X_1}{R_1}+\frac{X_2}{R_2}+\cdots\right)$	
2.4	$X_o = -\dfrac{R_oC_1P}{R_oC_1RCP^2+\left(\frac{1}{R}+\frac{1}{r_L}+\frac{1}{R_o}\right)R_oRCP+1}X_1$	



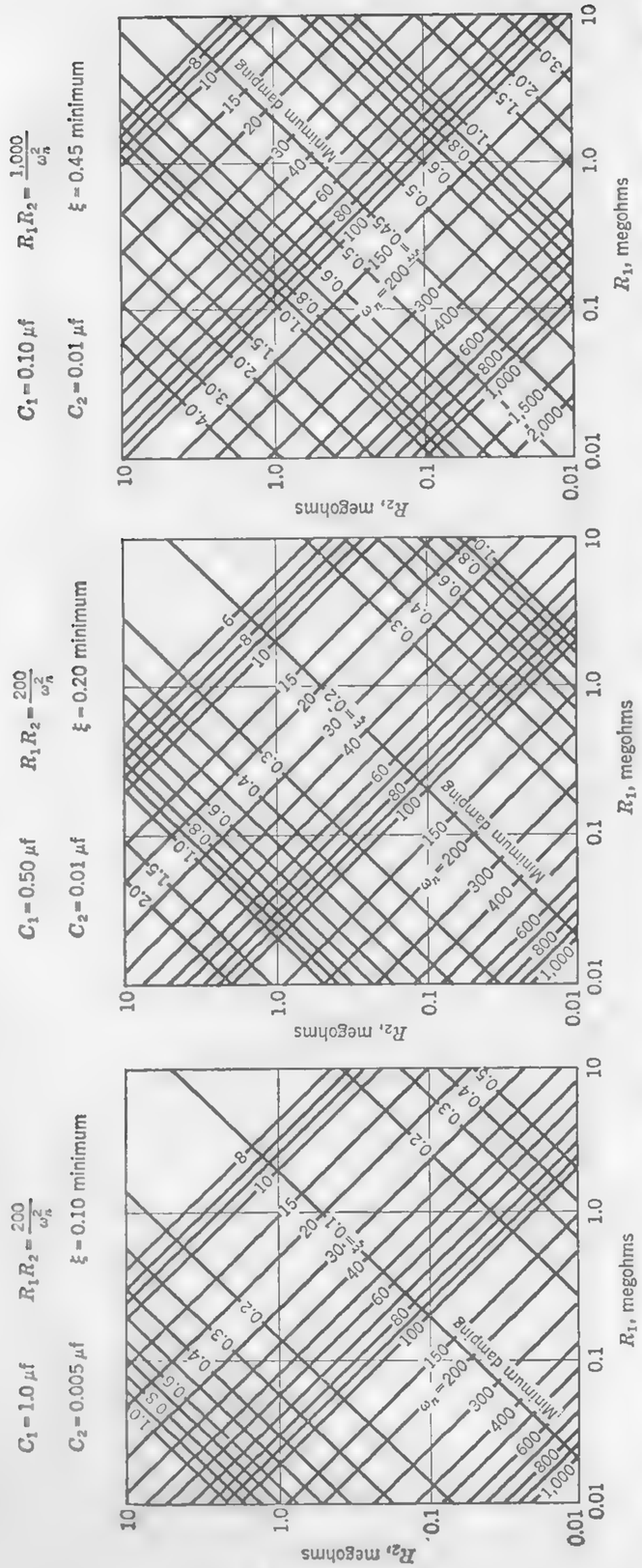

$C_1 = 0.10\ \mu f$ $R_1R_2 = \frac{1{,}000}{\omega_n^2}$ $\xi = 0.45$ minimum
$C_2 = 0.01\ \mu f$
Minimum damping
ω_n = 200
R_2, megohms
R_1, megohms
$C_1 = 0.50\ \mu f$ $R_1R_2 = \frac{200}{\omega_n^2}$ $\xi = 0.20$ minimum
$C_2 = 0.01\ \mu f$
$C_1 = 1.0\ \mu f$ $R_1R_2 = \frac{200}{\omega_n^2}$ $\xi = 0.10$ minimum
$C_2 = 0.005\ \mu f$

Fig. 2.2.3. Design nomographs for a simple circuit permitting economical computer representation of many mass-spring-dashpot systems (valves, rate gyros, accelerometers, etc.). (*From G. W. Smith and R. C. Wood, Principles of Analog Computation, McGraw-Hill Book Company, Inc., New York, 1959.*)

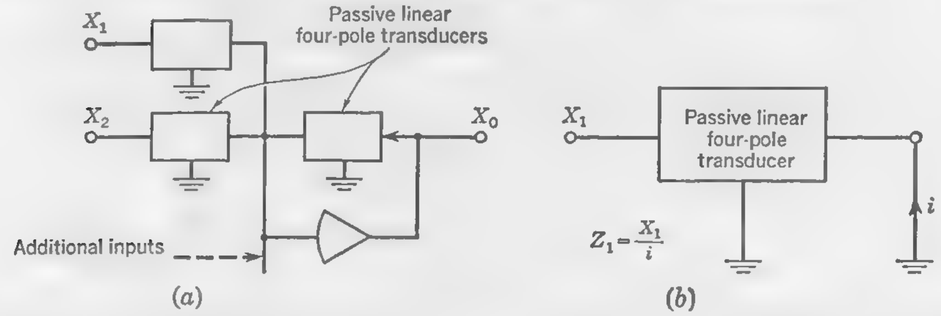


FIG. 2.2.4. Operational amplifier employing four-terminal input and feedback network (*a*), and definition of short-circuit transfer impedances (*b*). (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

2.2.6. Operational Amplifiers with Four-terminal Input and Feedback Networks.[1,2] If the input and feedback impedances of a parallel-feedback operational amplifier are replaced by linear four-terminal networks (Fig. 2.2.4*a*) Eq. (1) still describes the amplifier operation for high loop gain, provided that each impedance operator $Z_0(P)$, $Z_1(P)$, $Z_2(P)$, . . . is understood to denote the *short-circuit transfer impedance* (ratio of input voltage to short-circuit output current, Fig. 2.2.4*b*) of the four-terminal network in question. Table 2.2.4 lists transfer impedances of commonly used networks and is very useful for syntheses of operational amplifiers with specified transfer functions. The error analysis of Art. 2.2.2 does *not* apply; detailed performance analyses require solution of the node equations in each case.

2.2.7. Transfer-function Synthesis by Differential-analyzer Techniques.[1,2] Table 2.2.5 shows computer setups implementing commonly useful transfer functions with integrators, summers, and potentiometers by classical differential-analyzer techniques (Sec. 5.1). These circuits permit convenient adjustment of parameters and are suitable for use with "committed" operational amplifiers in commercial analog computers. Additional special circuits have been collected by Estes,[3] who points out that a circuit with the transfer function $(1 + T^2s^2)/(1 + Ts)^2$ (use circuit 8 of Table 2.2.5) is useful as a *notch filter* for rejecting an unwanted frequency component. Figure 2.2.5 shows a number of practical *differentiating circuits.*

ERRORS DUE TO PARASITIC CIRCUIT PARAMETERS

2.2.8. Errors Due to Parasitic Circuit Parameters. The design formulas of Arts. 2.2.1 to 2.2.7 are based on the assumption of *ideal resistances and capacitances* (and possibly inductances) in all operational-amplifier networks. In practice, the performance of operational amplifiers is compromised not only by insufficient loop gain (Art. 2.2.2) but also by *parasitic resistances, inductances, and capacitances* superimposed on the desired network elements. In particular, *integrator performance may suffer from resistor series inductance and parallel capacitance, and from integrating-capacitor leakage and dielectric absorption.* The relevant error analysis involves the composite design of operational amplifiers, coefficient potentiometers, patchbays, and cabling and is discussed in Sec. 2.4.

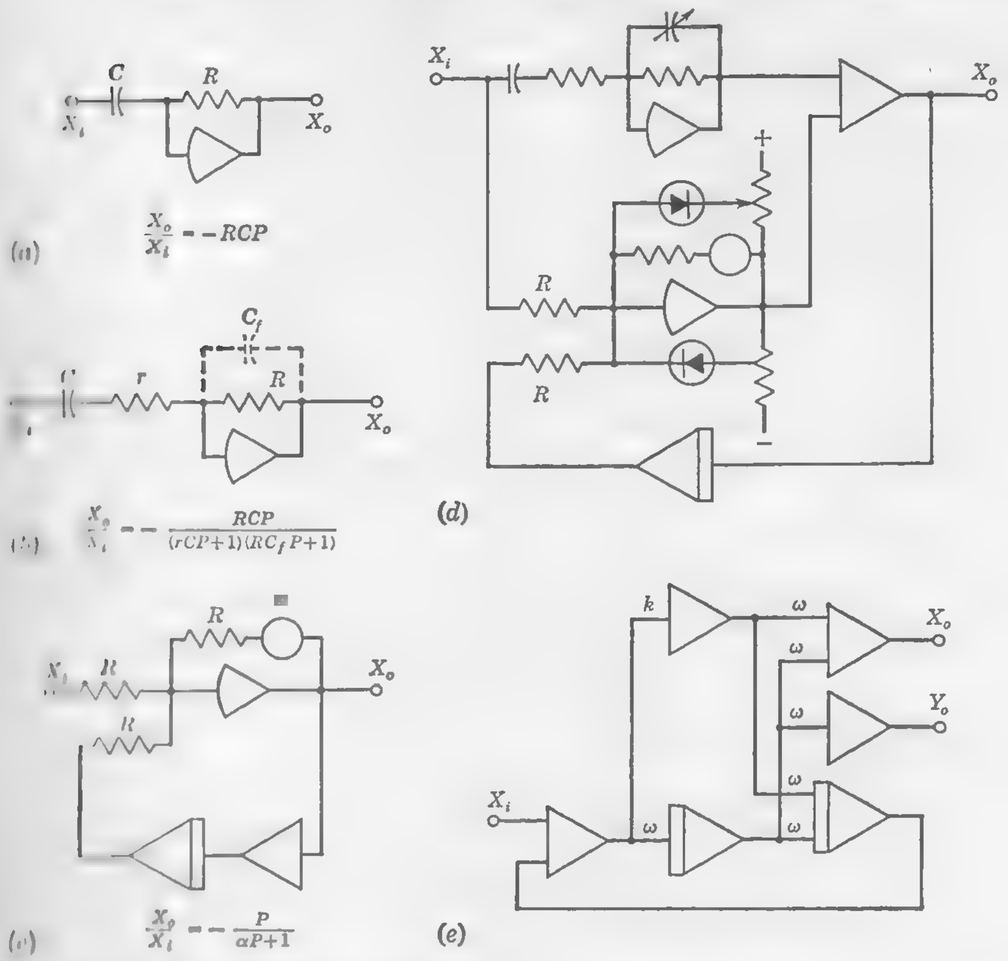


FIG. 2.2.5. Analog differentiating circuits.
FIG. 2.2.5*a*. Basic operational-amplifier differentiator.
FIG. 2.2.5*b*. Addition of the resistance r prevents overloads in the preceding amplifier and smooths noise peaks. Capacitor C_F may be added for additional smoothing.
FIG. 2.2.5*c*. Differentiation and smoothing by implicit computation. The high-gain amplifier may have to be feedback-limited (see also Fig. 2.2.5*d*).
FIG. 2.2.5*d*. Combination of Fig. 2.2.5*b* and *c*. The high-gain amplifier has adjustable gain and is feedback-limited (Sec. 3.3) to prevent overloads. (*Contributed by J. McLeod, [illegible] Laboratories.*)
FIG. 2.2.5*e*. This implicit-differentiation circuit yields two approximate-derivative outputs

$$X_0 = -\left(1 - \frac{P^2}{P^2 + k\omega P + \omega^2}\right) PX_1 \qquad Y_0 = -\left(1 - \frac{P^2 + k\omega P}{P^2 + k\omega P + \omega^2}\right) PX_1$$

X_0 is more accurate, and Y_0 is smoother. Integrators must be supplied with appropriate initial condition voltages. (*Contributed by L. Bauer, Reeves Instrument Corporation.*)

References

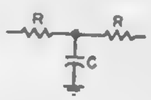
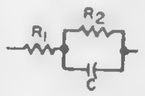
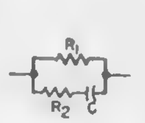
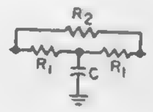
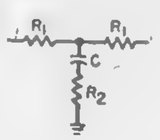
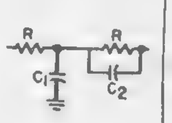
1. Korn, Granino A., and Theresa M. Korn: *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.
2. Booth, George W., and Roger C. Wood: *Principles of Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1959.
3. Estes, H. H., *et al.*: Techniques and Developments in the Analog-computer Field, *Rept. 6006*, Apr. 15, 1957, McDonnel Aircraft Corporation, St. Louis, Mo.

Table 2.2.4. Transfer Impedances for Operational Amplifiers*

This table is used to find networks whose transfer impedances $Z_o(P)$, $Z_1(P)$, $Z_2(P)$, $\cdots$, match the desired performance equation

$$X_o = -\frac{Z_o(P)}{Z_1(P)} X_1 \qquad \text{or} \qquad X_o = -Z_o(P)\left[\frac{X_1}{Z_1(P)} + \frac{X_2}{Z_2(P)} + \cdots\right]$$

for an operational amplifier of the type shown in Fig. 2.2.4.

Transfer-impedance function	Network	Relations	Inverse relations
A		$A = R$	$R = A$
$\dfrac{A}{1+PT}$		$A = R$ $T = RC$	$R = A$ $C = \dfrac{T}{A}$
$A(1+PT)$		$A = 2R$ $T = \dfrac{RC}{2}$	$R = \dfrac{A}{2}$ $C = \dfrac{4T}{A}$
$A\left(\dfrac{1+P\theta T}{1+PT}\right)$ $\theta < 1$		$A = R_1 + R_2$ $T = R_2C$ $\theta = \dfrac{R_1}{R_1+R_2}$	$R_1 = A\theta$ $R_2 = A(1-\theta)$ $C = \dfrac{T}{A(1-\theta)}$
		$A = R_1$ $T = (R_1+R_2)C$ $\theta = \dfrac{R_2}{R_1+R_2}$	$R_1 = A$ $R_2 = \dfrac{A\theta}{1-\theta}$ $C = \dfrac{T(1-\theta)}{A}$
$A\left(\dfrac{1+PT}{1+P\theta T}\right)$ $\theta < 1$		$A = \dfrac{2R_1R_2}{2R_1+R_2}$ $T = \dfrac{R_1C}{2}$ $\theta = \dfrac{2R_1}{2R_1+R_2}$	$R_1 = \dfrac{A}{2(1-\theta)}$ $R_2 = \dfrac{A}{\theta}$ $C = \dfrac{4T(1-\theta)}{A}$
		$A = 2R_1$ $T = \left(R_2 + \dfrac{R_1}{2}\right)C$ $\theta = \dfrac{2R_2}{2R_2+R_1}$	$R_1 = \dfrac{A}{2}$ $R_2 = \dfrac{A\theta}{4(1-\theta)}$ $C = \dfrac{4T(1-\theta)}{A}$
		$A = 2R$ $T = \dfrac{R}{2}(C_1+C_2)$ $\theta = \dfrac{2C_2}{C_1+C_2}$	$R = \dfrac{A}{2}$ $C_1 = \dfrac{2T(2-\theta)}{A}$ $C_2 = \dfrac{2T\theta}{A}$

* Reproduced by permission of the McGraw-Hill Publishing Company from F. R. Bradley and R. McCoy, Driftless D-C Amplifier, *Electronics*, April, 1952. This table was developed by S. Godet of the Reeves Instrument Corporation, New York City.

Table 2.2.4. Transfer Impedances for Operational Amplifiers (*Continued*)

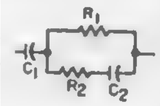
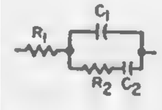
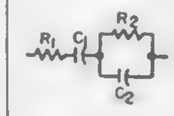
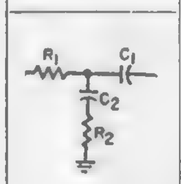
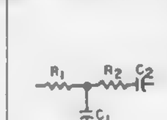
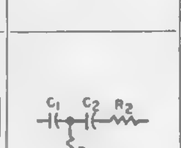
Transfer-impedance function	Network	Relations	Inverse relations
$\dfrac{1}{PT_1}\left[\dfrac{(1+PT_1)(1+PT_2)}{1+PT_3}\right]$ $T_1 < T_2 < T_3$		$B = C_1$ $T_2 = (R_1+R_2)C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \dfrac{T_1+T_3-T_2}{B}$ $R_2 = \dfrac{T_1T_3(T_1+T_3-T_2)}{B(T_3-T_2)(T_2-T_1)}$ $C_1 = B$ $C_2 = \dfrac{B(T_3-T_2)(T_2-T_1)}{(T_1+T_3-T_2)^2}$
		$B = C_1 + C_2$ $T_2 = R_2\left(\dfrac{C_1C_2}{C_1+C_2}\right)$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \dfrac{T_1T_2}{BT_2}$ $R_2 = \dfrac{(T_1T_2 + T_2T_3 - T_1T_3)^2}{BT_2(T_3-T_2)(T_2-T_1)}$ $C_1 = \dfrac{BT_2^2}{T_1T_2+T_2T_3-T_1T_3}$ $C_2 = \dfrac{B(T_3-T_2)(T_2-T_1)}{T_1T_2+T_2T_3-T_1T_3}$
		$B = C_1$ $T_2 = R_2C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = \dfrac{T_1T_2}{BT_2}$ $R_2 = \dfrac{(T_3-T_2)(T_2-T_1)}{BT_2}$ $C_1 = B$ $C_2 = \dfrac{BT_2^2}{(T_3-T_2)(T_2-T_1)}$
		$B = C_1$ $T_2 = R_2C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \dfrac{T_1T_2}{BT_2}$ $R_2 = \dfrac{T_1T_2T_3}{B(T_3-T_2)(T_2-T_1)}$ $C_1 = B$ $C_2 = \dfrac{B(T_3-T_2)(T_2-T_1)}{T_1T_3}$
$\dfrac{1}{PT_1}(1+PT_1)(1+PT_2)$ $T_1 \neq T_2$		$B = C_2$ $T_1T_2 = R_1R_2C_1C_2$ $T_1 + T_2 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \dfrac{(\sqrt{T_1}-\sqrt{T_2})^2}{B}$ $R_2 = \dfrac{\sqrt{T_1T_2}}{B}$ $C_1 = \dfrac{B\sqrt{T_1T_2}}{(\sqrt{T_1}-\sqrt{T_2})^2}$ $C_2 = B$
$\dfrac{1}{PT_1}\left[\dfrac{(1+PT_1)(1+PT_2)}{P\sqrt{T_1T_2}}\right]$ $T_1 \neq T_2$		$B = C_2$ $T_1T_2 = R_1R_2C_1C_2$ $T_1 + T_2 = R_1C_1 + R_2C_2 + R_1C_2$	$R = \dfrac{(\sqrt{T_1}-\sqrt{T_2})^2}{B}$ $R_2 = \dfrac{\sqrt{T_1T_2}}{B}$ $C_1 = \dfrac{B\sqrt{T_1T_2}}{(\sqrt{T_1}-\sqrt{T_2})^2}$ $C_2 = B$
$\dfrac{1}{PT_1}\left[\dfrac{(1+PT_1)(1+PT_2)}{P^2T_1T_2}\right]$ $T_1 < T_2$		$B = \dfrac{C_1C_2}{C_1+2C_2}$ $T_1 = RC_1$ $T_2 = R(C_1+2C_2)$	$R = \dfrac{T_1(T_2-T_1)}{2BT_2}$ $C_1 = \dfrac{2BT_2}{T_2-T_1}$ $C_2 = \dfrac{BT_2}{T_1}$

Table 2.2.4. Transfer Impedances for Operational Amplifiers (*Continued*)

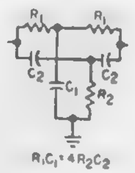
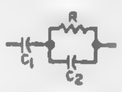
Transfer-impedance function	Network	Relations	Inverse relations
$A\left(\frac{1+PT_1}{1+P^2T_1T_2}\right)$		$A = 2R_1$ $T_1 = \frac{R_1C_1}{2} = 2R_2C_2$ $T_2 = R_1C_2$	$R_1 = \frac{A}{2}$ $R_2 = \frac{AT_1}{4T_2}$ $C_1 = \frac{4T_1}{A}$ $C_2 = \frac{2T_2}{A}$
$\frac{1}{PB}$		$B = C$	$C = B$
$\frac{1}{PB}(1+PT)$		$B = C$ $T = RC$	$R = \frac{T}{B}$ $C = B$
$\frac{1}{PB}\left(\frac{1+PT}{PT}\right)$		$B = \frac{C}{2}$ $T = 2RC$	$R = \frac{T}{4B}$ $C = 2B$
$\frac{1}{PB}\left(\frac{1+PT}{1+P\theta T}\right)$ $\theta < 1$		$B = C_1$ $T = R(C_1 + C_2)$ $\theta = \frac{C_2}{C_1 + C_2}$	$R = \frac{T(1-\theta)}{B}$ $C_1 = B$ $C_2 = \frac{B\theta}{1-\theta}$
		$B = C_1 + C_2$ $T = RC_2$ $\theta = \frac{C_1}{C_1 + C_2}$	$R = \frac{T}{B(1-\theta)}$ $C_1 = B\theta$ $C_2 = B(1-\theta)$
$\frac{1}{PB}\left[\frac{(1+PT_1)(1+PT_2)}{1+PT_3}\right]$ $T_1 < T_2 < T_3$		$B = C_1 + C_2$ $T_1 = R_1C_1$ $T_2 = (R_1 + R_2)\left(\frac{C_1C_2}{C_1 + C_2}\right)$ $T_3 = R_2C_2$	$R_1 = \frac{T_1(T_3 - T_1)}{B(T_2 - T_1)}$ $R_2 = \frac{T_3(T_3 - T_1)}{B(T_3 - T_2)}$ $C_1 = \frac{B(T_2 - T_1)}{T_3 - T_1}$ $C_2 = \frac{B(T_3 - T_2)}{T_3 - T_1}$
$A\left(\frac{1+PT_1}{1+PT_1+P^2T_1T_2}\right)$		$A = R_2$ $T_1 = 2R_1C$ $T_2 = \frac{R_2C}{2}$	$R_1 = \frac{AT_1}{4T_2}$ $R_2 = A$ $C = \frac{2T_2}{A}$
$A\left(\frac{1+PT_2}{1+PT_1+P^2T_1T_2}\right)$		$A = 2R$ $T_1 = 2RC_2$ $T_2 = \frac{RC_1}{2}$	$R = \frac{A}{2}$ $C_1 = \frac{4T_2}{A}$ $C_2 = \frac{T_1}{A}$

Table 2.2.4. Transfer Impedances for Operational Amplifiers (*Continued*)

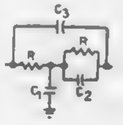
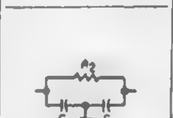
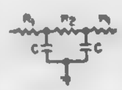
Transfer-impedance function	Network	Relations	Inverse relations
		$A = \frac{2R_1R_2}{(2R_1 + R_2)}$ $T_1 = \frac{R_1(R_1C_1 + 2R_2C_2)}{2R_1 + R_2}$ $T_2 = \frac{R_1R_2C_1C_2}{R_1C_1 + 2R_2C_2}$ $T_3 = \frac{R_1C_1}{2}$	$R_1 = \frac{AT_3^2}{2[T_3^2 - T_1(T_3 - T_2)]}$ $R_2 = \frac{AT_3^2}{T_1(T_3 - T_2)}$ $C_1 = \frac{4[T_3^2 - T_1(T_3 - T_2)]}{AT_3}$ $C_2 = \frac{T_1T_2}{AT_3}$
$A\left[\frac{1+PT_3}{1+PT_1+P^2T_1T_2}\right]$ $T_3 > \frac{T_1}{4}$ (complex roots) $T_3 > T_2$		$A = 2R_1$ $T_1 = R_2C_1 + 2R_1C_2$ $T_2 = \frac{R_1(R_1 + 2R_2)C_1C_2}{R_2C_1 + 2R_1C_2}$ $T_3 = \left(R_2 + \frac{R_1}{2}\right)C_1$	$R_1 = \frac{A}{2}$ $R_2 = \frac{AT_1(T_3 - T_2)}{4[T_3^2 - T_1(T_3 - T_2)]}$ $C_1 = \frac{4[T_3^2 - T_1(T_3 - T_2)]}{AT_3}$ $C_2 = \frac{T_1T_2}{AT_3}$
		$A = 2R$ $T_1 = R(C_3 + 2C_2)$ $T_2 = \frac{RC_3(C_1 + C_2)}{C_2 + 2C_3}$ $T_3 = \frac{R}{2}(C_1 + C_2)$	$R = \frac{A}{2}$ $C_1 = \frac{2[2T_3^2 - T_1(T_3 - T_2)]}{AT_3}$ $C_2 = \frac{2T_1(T_3 - T_2)}{AT_3}$ $C_3 = \frac{T_1T_2}{AT_3}$
$A\left[\frac{1+PT_3}{1+PT_1+P^2T_1T_2}\right]$ $T_3 > \frac{T_1}{4}$ (complex roots) $T_3 < T_1$		$A = R_2$ $T_1 = 2R_1C_1 + R_2C_2$ $T_2 = \frac{R_1R_2C_1(C_1 + 2C_2)}{2R_1C_1 + R_2C_2}$ $T_3 = 2R_1C_1$	$R_1 = \frac{AT_3^2}{4[T_1T_3 - T_3(T_1 - T_2)]}$ $R_2 = A$ $C_1 = \frac{2[T_1T_3 - T_3(T_1 - T_2)]}{AT_3}$ $C_2 = \frac{(T_1 - T_3)}{A}$
		$A = R_2$ $T_1 = \frac{C_1(2R_1C_2 + R_2C_1)}{2C_1 + C_2}$ $T_2 = \frac{R_1R_2C_1C_2}{2R_1C_2 + R_2C_1}$ $T_3 = \frac{2R_1C_1C_2}{(2C_1 + C_2)}$	$R_1 = \frac{AT_3^2}{4[T_1T_3 - T_3(T_1 - T_2)]}$ $R_2 = A$ $C_1 = \frac{2T_1T_2}{AT_3}$ $C_2 = \frac{4T_1T_2[T_1T_2 - T_3(T_1 - T_2)]}{AT_3^2(T_1 - T_2)}$
		$A = R_2$ $T_1 = \frac{R_1(2R_2 + R_3)C}{R_1 + R_2}$ $T_2 = \frac{R_2R_3C}{(2R_2 + R_3)}$ $T_3 = \frac{2R_1R_2C}{(R_1 + R_2)}$	$R_1 = \frac{AT_3^2}{2[2T_1T_2 - T_3(T_1 - T_2)]}$ $R_2 = \frac{AT_3}{2(T_1 - T_2)}$ $R_3 = A$ $C = \frac{2T_1T_2}{AT_3}$
$A(1+PT_1)(1+PT_3)$ $T_1 < T_3$		$A = 2R_1 + R_2$ $T_1 = \left(\frac{R_1R_2}{2R_1 + R_2}\right)C$ $T_2 = R_1C$	$R_1 = A\left(\frac{T_2 - T_1}{2T_2}\right)$ $R_2 = A\frac{T_1}{T_2}$ $C = \frac{2T_2^2}{A(T_2 - T_1)}$

Table 2.2.4. Transfer Impedances for Operational Amplifiers (*Continued*)

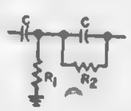
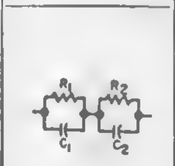
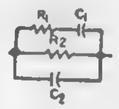
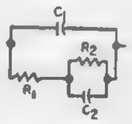
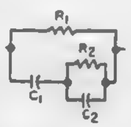
Transfer-impedance function	Network	Relations	Inverse relations
$\frac{1}{PB}\left(\frac{1+P\theta T}{1+PT}\right)$ $\theta < 1$		$B = C_2$ $T = RC_1\left(\frac{2C_2 + C_1}{C_2}\right)$ $\theta = \frac{2C_2}{2C_2 + C_1}$	$R = \frac{T\theta^2}{4B(1-\theta)}$ $C_1 = \frac{2B(1-\theta)}{\theta}$ $C_2 = B$
		$B = \frac{C_1^2}{2C_1 + C_2}$ $T = RC_2$ $\theta = \frac{2C_1}{2C_1 + C_2}$	$R = \frac{T\theta^2}{4B(1-\theta)}$ $C_1 = \frac{2B}{\theta}$ $C_2 = \frac{4B(1-\theta)}{\theta^2}$
		$B = \left(\frac{R_1}{R_1 + R_2}\right) C$ $T = R_2C$ $\theta = \frac{2R_1}{R_1 + R_2}$	$R_1 = \frac{T\theta^2}{2B(2-\theta)}$ $R_2 = \frac{T\theta}{2B}$ $C = \frac{2B}{\theta}$
$A\left[\frac{1+PT_2}{(1+PT_1)(1+PT_3)}\right]$ $T_1 < T_2 < T_3$		$A = R_1 + R_2$ $T_1 = R_1C_1$ $T_2 = \left(\frac{R_1R_2}{R_1+R_2}\right)(C_1+C_2)$ $T_3 = R_2C_2$	$R_1 = \frac{A(T_2 - T_1)}{T_3 - T_1}$ $R_2 = \frac{A(T_3 - T_2)}{T_3 - T_1}$ $C_1 = \frac{T_1(T_3 - T_1)}{A(T_2 - T_1)}$ $C_2 = \frac{T_3(T_3 - T_1)}{A(T_3 - T_2)}$
		$A = R_2$ $T_2 = R_1C_1$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = \frac{AT_2^2}{(T_3 - T_2)(T_2 - T_1)}$ $R_2 = A$ $C_1 = \frac{(T_3 - T_2)(T_2 - T_1)}{AT_2}$ $C_2 = \frac{T_1T_3}{AT_2}$
		$A = R_1 + R_2$ $T_2 = \left(\frac{R_1R_2}{R_1 + R_2}\right) C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = \frac{AT_2^2}{T_1T_2 + T_2T_3 - T_1T_3}$ $R_2 = \frac{A(T_3 - T_2)(T_2 - T_1)}{T_1T_2 + T_2T_3 - T_1T_3}$ $C_1 = \frac{T_1T_3}{AT_2}$ $C_2 = \frac{(T_1T_2 + T_2T_3 - T_1T_3)^2}{AT_2(T_3 - T_2)(T_2 - T_1)}$
		$A = R_1$ $T_2 = R_2(C_1 + C_2)$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = A$ $R_2 = \frac{A(T_3 - T_2)(T_2 - T_1)}{(T_1 + T_3 - T_2)^2}$ $C_1 = \frac{T_1 + T_3 - T_2}{A}$ $C_2 = \frac{T_1T_3(T_1 + T_3 - T_2)}{A(T_3 - T_2)(T_3 - T_1)}$

Table 2.2.4. Transfer Impedances for Operational Amplifiers (*Continued*)

Transfer-impedance function	Network	Relations	Inverse relations
$A\left[\frac{1+PT_2}{(1+PT_1)(1+PT_3)}\right]$ $T_3 \leq T_1 \leq T_2$		$A = 2R_1 + \frac{R_1^2}{R_2}$ $T_1 = R_1C_1$ $T_2 = \left(\frac{R_1R_2}{R_1+2R_2}\right)(C_1+C_2)$ $T_3 = R_1C_2$	$R_1 = \frac{AT_2}{(T_1 + T_3)}$ $R_2 = \frac{AT_2^2}{(T_1 + T_3)(T_1 + T_3 - 2T_2)}$ $C_1 = \frac{T_1(T_1 + T_3)}{AT_2}$ $C_2 = \frac{T_3(T_1 + T_3)}{AT_2}$
$A\left[\frac{1+PT_2}{(1+PT_1)(1+PT_3)}\right]$ $T_1 \leq T_3 \leq T_2$		$A = R_1 + R_2$ $T_1 = R_1C_1$ $T_2 = \frac{R_1R_2}{R_1 + R_2}(2C_1 + C_2)$ $T_3 = R_2C_1$	$R_1 = \frac{AT_1}{(T_1 + T_3)}$ $R_2 = \frac{AT_3}{(T_1 + T_3)}$ $C_1 = \frac{(T_1 + T_3)}{A}$ $C_2 = \frac{(T_1 + T_3)}{A}\left(\frac{T_2}{T_3} + \frac{T_2}{T_1} - 2\right)$

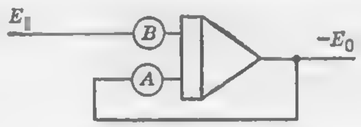
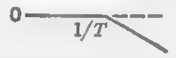
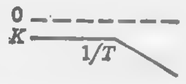
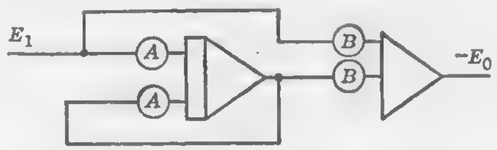
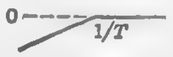
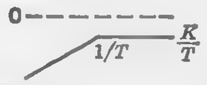
Table 2.2.5. Differential-analyzer Representation of Transfer Functions*

The table contains computer circuits for the simulation of all possible transfer functions which may be represented by any combination of three-terminal, two-, three-, four-, or five-element RC networks (see also Table 2.2.1). Note that $T_1 > T_2 > T_3 > T_4$ in each case.

Included with the circuits are the associated Bode plots, factored transfer functions, equations for transfer-function time constants and gains in terms of analog gains and equations for the necessary analog-circuit gains as functions of the desired time constants.

If there are more analog-circuit gains to be determined from time-constant relations than there are independent equations, assume one of the circuit gains and solve for those remaining from the equations given. In such cases, equations for the gains assumed are omitted from the tabulation of equations for analog-circuit gains in terms of transfer-function time constants.

All analog-circuit gains are shown on the diagrams as potentiometer settings. Where a circuit gain is greater than 1, a gain greater than 1 must be used on the associated amplifier.

No.	Bode plot	Transfer function	Time constants	Gains
				
1		$\frac{1}{Ts+1}$	$T=\frac{1}{A}$	$A=B=\frac{1}{T}$
2		$\frac{K}{Ts+1}$	$T=\frac{1}{A}$ $K=\frac{B}{A}$	$A=\frac{1}{T}$ $B=AK$
				
3		$\frac{Ts}{Ts+1}$	$T=\frac{1}{A}$	$A=\frac{1}{T}$ $B=1$
4		$\frac{Ks}{Ts+1}$	$T=\frac{1}{A}$ $K=BT$	$A=\frac{1}{T}$ $B=AK$

* Heizer, L. E., and S. J. Abraham, Transfer-function Simulation by Means of Amplifiers and Potentiometers, *CONVAIR-Fort Worth Rept.* Sept. 16, 1955, CONVAIR (a division of General Dynamics Corporation), Fort Worth, Tex.; reprinted by permission.

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

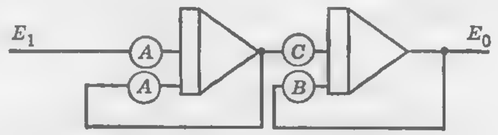
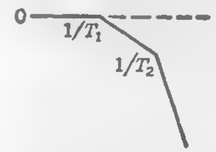
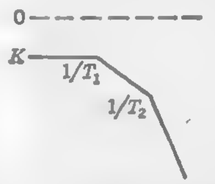
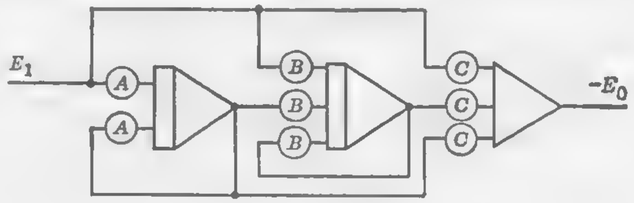
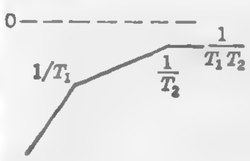
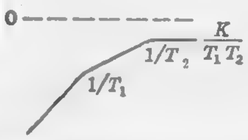
No.	Bode plot	Transfer function	Time constants	Gains
				
5		$\frac{1}{(T_1s+1)(T_2s+1)}$	$T_1=\frac{1}{A}$ $T_2=\frac{1}{B}$	$A=\frac{1}{T_1}$ $B=\frac{1}{T_2}$ $C=B$
6		$\frac{K}{(T_1s+1)(T_2s+1)}$	$T_1=\frac{1}{A}$ $T_2=\frac{1}{B}$	$A=\frac{1}{T_1}$ $B=\frac{1}{T_2}$ $C=\frac{K}{T_2}$
				
7		$\frac{s^2}{(T_1s+1)(T_2s+1)}$	$T_1=\frac{1}{A}$ $T_2=\frac{1}{B}$	$A=\frac{1}{T_1}$ $B=\frac{1}{T_2}$ $C=\frac{1}{T_1T_2}$
8		$\frac{Ks^2}{(T_1s+1)(T_2s+1)}$	$T_1=\frac{1}{A}$ $T_2=\frac{1}{B}$ $K=\frac{C}{AB}$	$A=\frac{1}{T_1}$ $B=\frac{1}{T_2}$ $C=\frac{K}{T_1T_2}$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

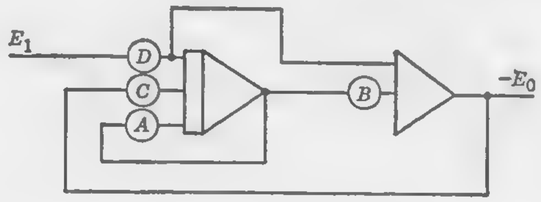
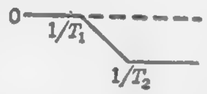
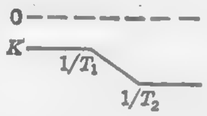
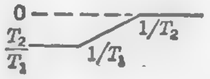
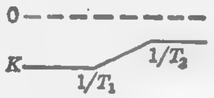
No.	Bode plot	Transfer function	Time constants	Gains
				
9		$\frac{T_2 s + 1}{T_1 s + 1}$	$T_1 = \frac{1}{A - BC}$ $T_2 = \frac{1}{A - B}$	$A = B + \frac{1}{T_2}$ $C = \frac{T_1 - T_2}{BT_1T_2} + 1$ $D = \frac{T_2}{T_1}$
10		$\frac{K(T_2 s + 1)}{T_1 s + 1}$	$T_1 = \frac{1}{A - BC}$ $T_2 = \frac{1}{A - B}$ $K = \frac{T_1 D}{T_2}$	$A = B + \frac{1}{T_2}$ $C = \frac{T_1 - T_2}{BT_1T_2} + 1$ $D = \frac{KT_2}{T_1}$
11		$\frac{K(T_1 s + 1)}{T_2 s + 1}$	$T_1 = \frac{1}{A - B}$ $T_2 = \frac{1}{A}$ $K = \frac{T_2}{T_1}$	$A = \frac{1}{T_2}$ $B = \frac{T_2 - T_1}{T_1T_2}$ $C = 0$ $D = 1$
12		$\frac{K(T_1 s + 1)}{T_2 s + 1}$	$T_1 = \frac{1}{A - B}$ $T_2 = \frac{1}{A}$ $K = \frac{T_2 D}{T_1}$	$A = \frac{1}{T_2}$ $B = \frac{T_2 - T_1}{T_1T_2}$ $C = 0$ $D = \frac{KT_1}{T_2}$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

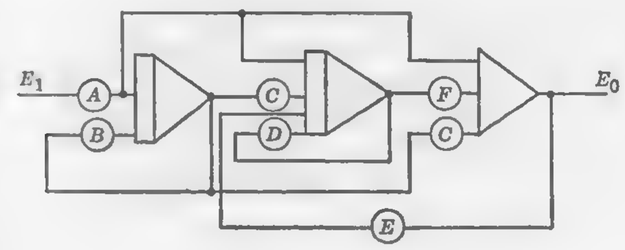
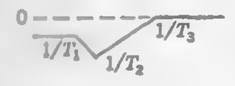
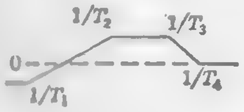
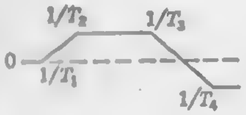
No.	Bode plot	Transfer function	Time constants	Gains
				
10		$\frac{K(T_2 s + 1)^2}{(T_1 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{B}$ $T_2 = \frac{1}{B - C} = \frac{1}{D - F}$ $T_1 = \frac{1}{D - EF}$ $K = \frac{T_1 T_2}{(T_2)^2}$	$A = 1$ $B = \frac{1}{T_2}$ $B - C = D - F = \frac{1}{T_2}$ $D - EF = \frac{1}{T_1}$
11		$\frac{K(T_1 s + 1)(T_4 s + 1)}{(T_2 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{B - C}$ $T_2 = \frac{1}{B}$ $T_3 = \frac{1}{D - EF}$ $T_4 = \frac{1}{D - F}$ $K = \frac{T_2 T_3}{T_1 T_4}$	$A = 1$ $B = \frac{1}{T_2}$ $B - C = \frac{1}{T_1}$ $D - F = \frac{1}{T_4}$ $D - EF = \frac{1}{T_3}$
12		$\frac{(T_1 s + 1)(T_4 s + 1)}{(T_2 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{B - C}$ $T_2 = \frac{1}{B}$ $T_3 = \frac{1}{D - EF}$ $T_4 = \frac{1}{D - F}$	$A = \frac{T_1 T_4}{T_2 T_3}$ $B = \frac{1}{T_2}$ $B - C = \frac{1}{T_1}$ $D - F = \frac{1}{T_4}$ $D - EF = \frac{1}{T_3}$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

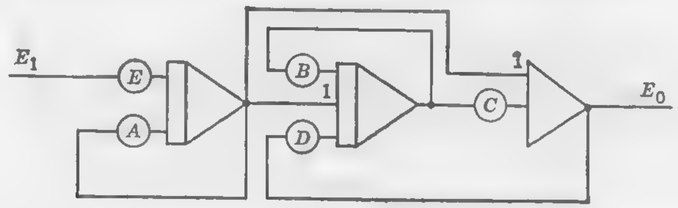
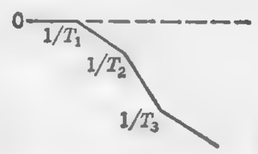
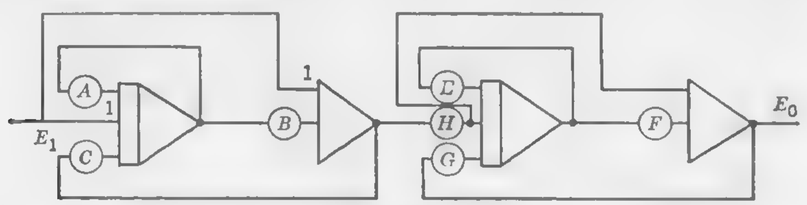
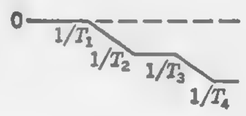
No.	Bode plot	Transfer function	Time constants	Gains
				
16		$\frac{T_2 s + 1}{(T_1 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{A}$ $T_2 = \frac{1}{B - CD}$ $T_3 = \frac{1}{B - C}$	$A = \frac{1}{T_1}$ $B = C + \frac{1}{T_3}$ $D = \frac{1}{CT_3} - \frac{1}{CT_2} + 1$ $E = \frac{T_3}{T_1 T_2}$
				
17		$\frac{(T_2 s + 1)(T_4 s + 1)}{(T_1 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{A - BC}$ $T_2 = \frac{1}{A - B}$ $T_3 = \frac{1}{E - FG}$ $T_4 = \frac{1}{E - F}$	$A = B + \frac{1}{T_2}$ $C = \frac{T_1 - T_3}{BT_1 T_2} + 1$ $E = F + \frac{1}{T_4}$ $G = \frac{T_2 - T_4}{FT_3 T_4}$ $H = \frac{T_2 T_4}{T_1 T_3}$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

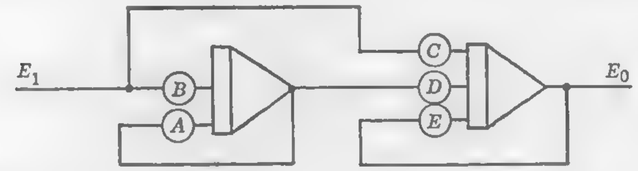
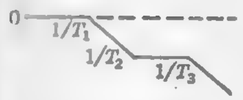
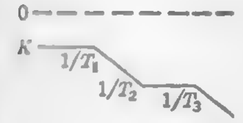
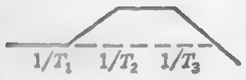
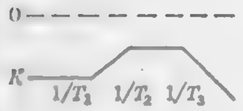
No.	Bode plot	Transfer function	Time constants	Gains
				
		$\frac{T_2 s + 1}{(T_1 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{E}$ $T_2 = \frac{E}{EA - D}$ $T_3 = \frac{1}{A}$	$A = \frac{1}{T_3}$ $B = AT_2$ $C = BE$ $D = EA - \frac{E}{T_2}$ $E = \frac{1}{T_1}$
		$\frac{K(T_2 s + 1)}{(T_1 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{E}$ $T_2 = \frac{E}{EA - D}$ $T_3 = \frac{1}{A}$ $K = \frac{T_2 B}{T_2}$	$A = \frac{1}{T_3}$ $B = KAT_2$ $C = BE$ $D = EA - \frac{E}{T_2}$ $E = \frac{1}{T_1}$
		$\frac{T_1 s + 1}{(T_2 s + 1)(T_3 s + 1)}$	$T_1 = \frac{E}{EA - D}$ $T_2 = \frac{1}{E}$ $T_3 = \frac{1}{A}$	$A = \frac{1}{T_3}$ $B = AT_1$ $C = BE$ $D = EA - \frac{E}{T_1}$ $E = \frac{1}{T_2}$
		$\frac{K(T_1 s + 1)}{(T_2 s + 1)(T_3 s + 1)}$	$T_1 = \frac{E}{EA - D}$ $T_2 = \frac{1}{E}$ $T_3 = \frac{1}{A}$ $K = \frac{T_2 B}{T_1}$	$A = \frac{1}{T_3}$ $B = KAT_1$ $C = BE$ $D = EA - \frac{E}{T_1}$ $E = \frac{1}{T_2}$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

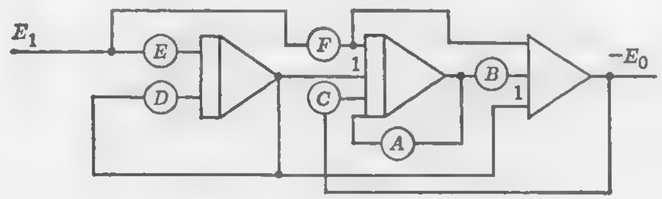
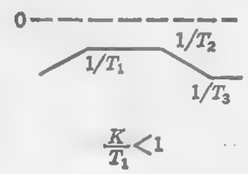
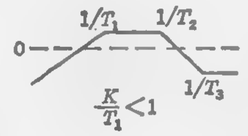
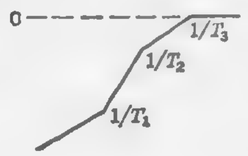
No.	Bode plot	Transfer function	Time constants	Gains
				
22		$\frac{Ks(T_2s+1)}{(T_1s+1)(T_2s+1)}$	$T_1=\frac{1}{D}$ $T_2=\frac{1}{A-BC}$ $T_3=\frac{1}{A-B}$ $K=\frac{T_2FT_1}{T_3}$	$A=B+\frac{1}{T_3}$ $C=\frac{T_2-T_3}{BT_2T_3}+1$ $D=\frac{1}{T_1}$ $E=DF$ $F=\frac{T_3K}{T_2T_1}$
23		$\frac{Ks(T_2s+1)}{(1+T_1s)(1+T_2s)}$	$T_1=\frac{1}{D}$ $T_2=\frac{1}{A-BC}$ $T_3=\frac{1}{A-B}$ $K=\frac{T_2T_1F}{T_3}$	$A=B+\frac{1}{T_3}$ $C=\frac{T_2-T_3}{BT_2T_3}+1$ $D=\frac{1}{T_1}$ $E=DF$ $F=\frac{T_3K}{T_2T_1}$
24		$\frac{Ks(T_1s+1)}{(T_2s+1)(T_3s+1)}$	$T_1=\frac{1}{A-B}$ $T_2=\frac{1}{A}$ $T_3=\frac{1}{D}$	$A=\frac{1}{T_2}$ $B=\frac{T_2-T_1}{T_1T_2}$ $D=\frac{1}{T_3}$ $E=F=\frac{T_1K}{T_2T_3}=1$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

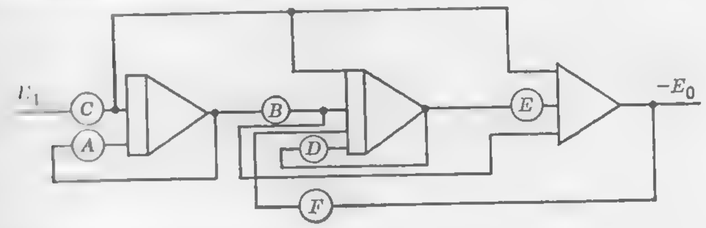
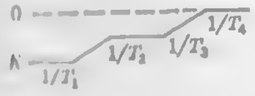
No.	Bode plot	Transfer function	Time constants	Gains
				
		$\frac{(T_2s+1)(T_3s+1)}{(T_1s+1)(T_4s+1)}$	$T_1=\frac{1}{D-EF}$ $T_2=\frac{1}{D-E}$ $T_3=\frac{1}{A-B}$ $T_4=\frac{1}{A}$	$A=\frac{1}{T_4}$ $B=\frac{T_3-T_4}{T_3T_4}$ $C=\frac{T_2T_3}{T_1T_4}=1$ $D=E+\frac{1}{T_2}$ $F=\frac{T_1-T_2}{ET_1T_2}+1$
		$\frac{(T_2s+1)^2}{(T_1s+1)(T_3s+1)}$	$T_1=\frac{1}{D-EF}$ $T_2=\frac{1}{D-E}$ $T_3=\frac{1}{A-B}$ $T_4=\frac{1}{A}$	$A=\frac{1}{T_3}$ $B=\frac{T_2-T_3}{T_2T_3}$ $C=1$ $D=E+\frac{1}{T_2}$ $F=\frac{T_1-T_2}{ET_1T_2}+1$
		$\frac{K(T_1s+1)(T_3s+1)}{(T_2s+1)(T_4s+1)}$	$T_1=\frac{1}{D-E}$ $T_2=\frac{1}{D}$ $T_3=\frac{1}{A-B}$ $T_4=\frac{1}{A}$ $K=\frac{T_2T_4}{T_1T_3}$	$A=\frac{1}{T_4}$ $B=\frac{T_3-T_4}{T_3T_4}$ $C=1$ $D=\frac{1}{T_2}$ $E=\frac{T_1-T_2}{T_1T_2}$ $F=0$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

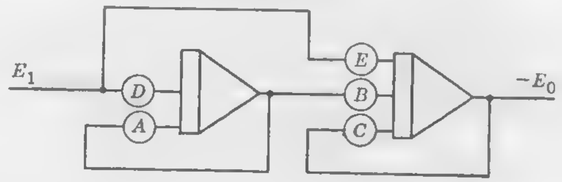
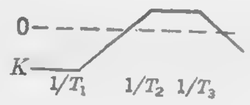
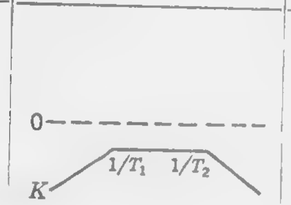
No.	Bode plot	Transfer function	Time constants	Gains
				
28		$\frac{K(T_1 s + 1)}{(T_2 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{A - B}$ $T_2 = \frac{1}{A}$ $T_3 = \frac{1}{C}$ $K = \frac{DT_2T_3}{T_1}$	$A = \frac{1}{T_2}$ $B = \frac{T_1 - T_2}{T_1T_2}$ $C = \frac{1}{T_3}$ $D = E = \frac{KT_1}{T_2T_3}$
29		$\frac{Ks}{(T_1 s + 1)(T_2 s + 1)}$	$T_1 = \frac{1}{A} = \frac{K}{D}$ $T_2 = \frac{1}{B} = \frac{1}{C}$ $K = ET_2T_1$	$A = \frac{1}{T_1}$ $B = C = \frac{1}{T_2}$ $D = \frac{K}{T_1^2}$ $E = \frac{K}{T_2T_1}$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

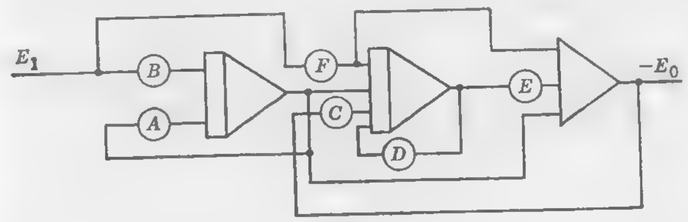
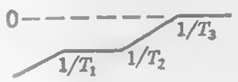
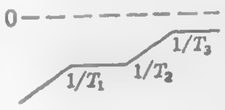
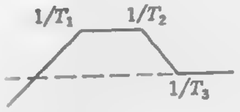
No.	Bode plot	Transfer function	Time constants	Gains
				
30		$\frac{Ks(T_2 s + 1)}{(T_1 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{A}$ $T_2 = \frac{1}{D - E}$ $T_3 = \frac{1}{D}$ $K = \frac{T_3T_1}{T_2}$	$A = \frac{1}{T_1}$ $B = AF$ $C = 0$ $D = \frac{1}{T_3}$ $E = \frac{T_2 - T_3}{T_2T_3}$
31		$\frac{Ks(T_2 s + 1)}{(T_1 s + 1)(T_3 s + 1)}$	$T_1 = \frac{1}{A}$ $T_2 = \frac{1}{D - E}$ $T_3 = \frac{1}{D}$ $K = \frac{T_3FT_1}{T_2}$	$A = \frac{1}{T_1}$ $B = AF$ $C = 0$ $D = \frac{1}{T_3}$ $E = \frac{T_2 - T_3}{T_2T_3}$ $F = \frac{T_2K}{T_3T_1}$
32		$\frac{Ks(T_3 s + 1)}{(T_1 s + 1)(T_2 s + 1)}$	$T_1 = \frac{1}{A}$ $T_2 = \frac{1}{D - EC}$ $T_3 = \frac{1}{D - E}$ $K = \frac{T_2FT}{T_3}$	$A = \frac{1}{T_1}$ $B = AF$ $C = \frac{T_2T_3}{ET_2T_3}$ $D = E + \frac{1}{T_3}$ $F = \frac{T_3K}{T_1T_2} = 1$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

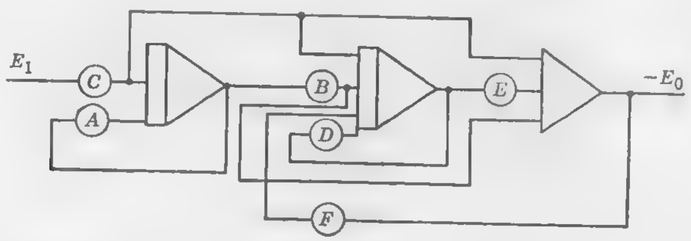
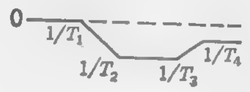
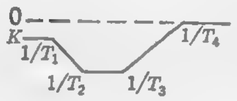
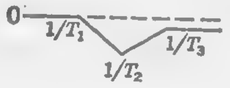
No.	Bode plot	Transfer function	Time constants	Gains
				
33		$\frac{(T_2s+1)(T_3s+1)}{(T_1s+1)(T_4s+1)}$	$T_1 = \frac{1}{D-EF}$ $T_2 = \frac{1}{D-E}$ $T_3 = \frac{1}{A-B}$ $T_4 = \frac{1}{A}$	$A = \frac{1}{T_4}$ $B = \frac{T_3 - T_4}{T_3T_4}$ $C = \frac{T_2T_3}{T_1T_4}$ $D = E + \frac{1}{T_2}$ $F = \frac{T_1 - T_2}{ET_1T_2} + 1$
34		$\frac{K(T_2s+1)(T_3s+1)}{(T_1s+1)(T_4s+1)}$	$T_1 = \frac{1}{D-EF}$ $T_2 = \frac{1}{D-E}$ $T_3 = \frac{1}{A-B}$ $T_4 = \frac{1}{A}$ $K = \frac{T_1T_4}{T_2T_3}$	$A = \frac{1}{T_4}$ $B = \frac{T_3 - T_4}{T_3T_4}$ $C = 1$ $D = E + \frac{1}{T_2}$ $F = \frac{T_1 - T_2}{ET_1T_2} + 1$
35		$\frac{(T_2s+1)^2}{(T_1s+1)(T_3s+1)}$	$T_1 = \frac{1}{D-EF}$ $T_2 = \frac{1}{D-E}$ $= \frac{1}{A-B}$ $T_3 = \frac{1}{A}$ $(T_2)^2 < T_2T_3$	$A = \frac{1}{T_3}$ $B = \frac{T_2 - T_3}{T_2T_3}$ $C = \frac{(T_2)^2}{T_1T_3}$ $D = E + \frac{1}{T_2}$ $F = \frac{T_1 - T_2}{ET_1T_2} + 1$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

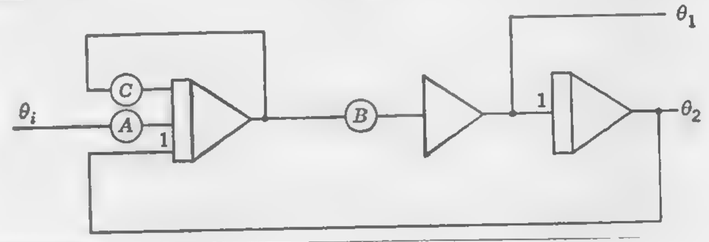
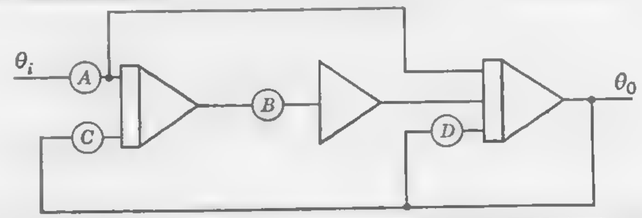
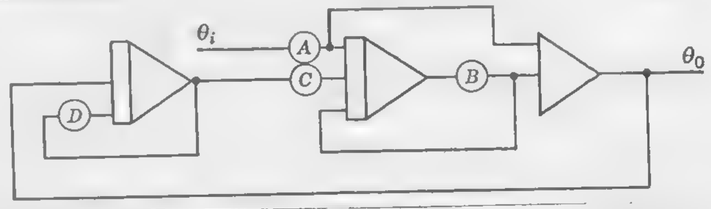
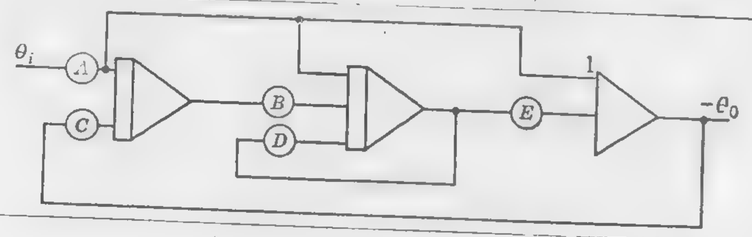
No.	Transfer function	Time constants	Gains
			
36	$\frac{\theta_1}{\theta_i} = \frac{Ks}{T^2s^2 + 2hTs + 1}$ $\frac{\theta_2}{\theta_i} = \frac{-K}{T^2s^2 + 2hTs + 1}$	$K = A$ $T = \frac{1}{\sqrt{B}}$ $h = \frac{C}{2\sqrt{B}}$	$A = K$ $B = \frac{1}{T^2}$ $C = \frac{2h}{T}$
			
37	$\frac{\theta_0}{\theta_i} = \frac{-K(T_1s+1)}{T_2^2s^2 + 2hT_2s + 1}$	$K = \frac{A}{C}$ $T_1 = \frac{1}{B}$ $T_2 = \frac{1}{\sqrt{BC}}$ $h = \frac{DT_2}{2}$	$A = CK$ $B = \frac{1}{T_1}$ $C = \frac{T_1}{T_2^2}$ $D = \frac{2h}{T_2}$
			
38	$\frac{\theta_0}{\theta_i} = \frac{Ks(T_1s+1)}{T_2^2s^2 + 2hT_2s + 1}$	$K = \frac{AD}{B(D+C)}$ $T_1 = \frac{1}{D}$ $T_2 = \frac{1}{\sqrt{B(D+C)}}$ $h = \frac{B+D}{2\sqrt{B(D+C)}}$	$A = \frac{KB(D+C)}{D}$ $B = \frac{2h}{T_2} - \frac{1}{T_1}$ $C = \frac{T_2^2 + T_1^2 - 2hT_1T_2}{T_1T_2(2hT_1 - T_2)}$ $D = \frac{1}{T_1}$

Table 2.2.5. Differential-analyzer Representation of Transfer Functions (*Continued*)

No.	Transfer function	Time constants	Gains
39	 $\frac{\theta_0}{\theta_i} = \frac{-K(T_1^2 s^2 + 2h_1 T_1 s + 1)}{T_2^2 s^2 + 2h_2 T_2 s + 1}$	$K = \frac{A}{C}$ $T_1 = \frac{1}{\sqrt{BE}}$ $h_1 = \frac{D - E}{2\sqrt{BE}}$ $T_2 = \frac{1}{\sqrt{BCE}}$ $h_2 = \frac{D}{2\sqrt{BCE}}$	$A = CK$ $B = \frac{T_2}{2T_1(h_2 T_1 - h_1 T_2)}$ $C = \frac{T_1^2}{T_2^2}$ $D = \frac{2h_2}{T_2}$ $E = 2\left(\frac{h_2}{T_2} - \frac{h_1}{T_1}\right)$

PART 3: DESIGN OF VACUUM-TUBE D-C AMPLIFIERS

By EDWARD M. BILLINGHURST

DESIGN SPECIFICATIONS

2.3.1. Typical Requirements. Operational amplifiers (Sec. 2.2) have primary specifications based on the required computing performance and secondary specifications such as drift stability, cost, and component availability.. Typical requirements follow:

	Low-cost d-c analog computer	High-quality d-c analog computer
Output voltage range (100 cps)	±50 or ±100 volts	±100 volts
Output current (100 cps)	3 to 15 ma	20 to 30 ma
Open-loop gain at d-c	500 to 10^5	10^7 to 10^8
Open-loop gain at 100 cps	500 to 10^4	10^4
Closed-loop phase shift at 100 cps (unity gain)	Less than 0.5 deg	Less than 0.1 deg
Closed-loop phase shift at 100 cps (gain of 10)	Less than 1 deg	Less than 0.15 deg
D-C offset (referred to input)	10 to 50 mv	Less than 100 μv
8-hr drift (referred to input)	10 mv	20 to 100 μv
Noise	100 mv peak-to-peak	10 mv peak-to-peak

Examples of computer-amplifier design will be found in Sec. 2.5.

2.3.2. Operational-amplifier Performance vs. Open-loop Gain and Phase Shift (see also Sec. 2.2). In Fig. 2.3.1*a*, the output voltage E_o of a representative operational amplifier is given by

$$E_o = -Z_f \sum_{i=1}^{n} \frac{E_i}{Z_i} \left[\frac{\frac{AY_oY_f}{Y_aY_b - Y_f^2}}{1 + \frac{AY_oY_f}{Y_aY_b - Y_f^2}} \right] \left[1 - \frac{Y_f}{AY_o} \right] \tag{2.3.1}$$

where the open-loop gain $(-A)$ as well as the various impedances and admittances Z and Y are functions of the complex-frequency variable s, and

$$Y_k = \frac{1}{Z_k} \tag{2.3.2}$$

Y_a is the admittance of the parallel combination of Z_f, Z_g, and $Z_1, \ldots, Z_n$ (Fig. 2.3.1*b*).

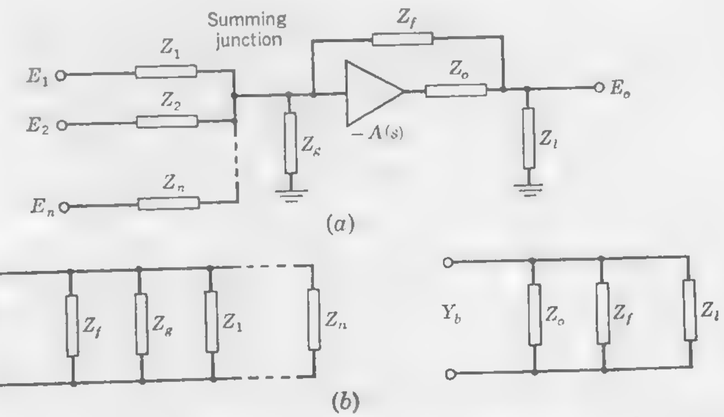


FIG. 2.3.1. Typical operational-amplifier system block diagram (*a*), and networks forming Y_a and Y_b(*b*).

Y_b is the admittance of the parallel combination of Z_o, Z_f, and Z_l (Fig. 2.3.1*b*); Z_o is the output impedance of the final stage of the amplifier.

The first term on the right of Eq. (1) represents *the desired performance equation of the operational amplifier*. The second term determines *stability* and *low-frequency errors*. The third term is usually of interest only at frequencies above the working frequency range of the computer. It does not generally contribute to computing errors or amplifier stability. Y_a and Y_b (Fig. 2.3.1*b*) usually involve only resistances and capacitances and can often be reduced to a single R and C in parallel.

Considering the second term in Eq. (1), the effect of the external circuitry is to contribute additional attenuation and phase-shift terms to the amplifier gain, $[-A(s)]$. The Nyquist stability criteria[1] require that the phase shift of

$$\frac{A(s)Y_oY_f}{Y_aY_b - Y_f^2} \tag{2.3.3}$$

be less than 180 deg when its magnitude exceeds unity. It is desirable to design $A(s)$ so that the phase shift is less than 90 deg for $|A| \geq 1$ (Art. 2.3.10). For the time being, assume that $|A(j\omega)| \geq 1$ at frequencies of $f = \omega/2\pi$ below about 1 Mc.

Figure 2.3.2 illustrates a typical shape of the frequency response $[A(j\omega)$ vs. $\omega]$. Large low-frequency gain reduces computing errors. To determine the required low-

frequency gain, approximate Eq. (1) by

$$E_o = -Z_f \sum_{i=1}^{n} \frac{E_i}{Z_i} \left[\frac{\dfrac{A(j\omega)}{1 + Z_f Y_a}}{1 + \dfrac{A(j\omega)}{1 + Z_f Y_a}} \right] \approx -Z_f \sum_{i=1}^{n} \frac{E_i}{Z_i} \left[1 - \frac{1 + Z_f Y_a}{A(j\omega)} \right] \quad (2.3.4)$$

The bracketed term determines the amplitude error at a particular frequency. The low-frequency amplitude error is given by

$$\epsilon = \left[\frac{\dfrac{A(j\omega)}{1 + Z_f Y_a}}{1 + \dfrac{A(j\omega)}{1 + Z_f Y_a}} \right] \quad (2.3.5)$$

Note that this is the error introduced by finite amplifier gain. Additional magnitude and phase errors may be included in the first term of Eq. (1) to show the effect of design tolerances in Z_i and Z_f.

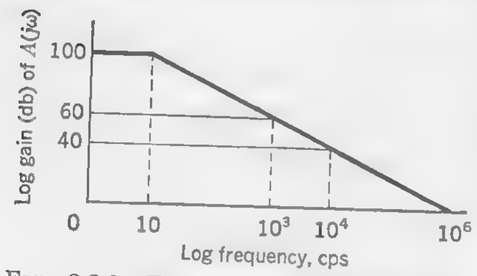


FIG. 2.3.2. Typical open-loop gain-frequency characteristic of an operational amplifier.

For low-gain amplifiers, $A(j\omega)$ will have low phase shift in the computing range, while high-gain amplifiers may have 90 deg phase shift near the upper limit of the computing range (e.g., at 100 cps). Two examples will serve to illustrate the method of determining the required low-frequency gain.

Example 1. *Low-cost Summing Amplifier.* Assume that Z_f and $Z_1, \ldots, Z_n$ are 1-megohm resistors; $Z_g = 1$ megohm at computing frequencies; a maximum of two input resistors is required, so that $1/Y_a = 250$ kilohms; 1 per cent amplitude accuracy is required. For this low accuracy, it will initially be assumed that $A(j\omega)$ has zero phase shift at 100 cps. The required condition is then

$$\left| \frac{\dfrac{A(j\omega)}{1 + Z_f Y_a}}{1 + \dfrac{A(j\omega)}{1 + Z_f Y_a}} \right| \geq 0.99 \quad (2.3.6)$$

From this, $A(j\omega)$ at 100 cps must be 495, or about 54 db. From Fig. 2.3.2, we see that this much gain requires a response with a corner frequency at 1 kc, so that $A(j\omega)$ will have low phase shift at 100 cps. Hence the amplifier would be designed to have a gain of at least 500 at low frequencies. If the amplifier is to be used as an integrator (i.e., if Z_f is a capacitor), the gain may be recalculated on this basis to determine whether it is satisfactory.

Example 2. *High-quality Summing Amplifier.* Assume the same input and feedback resistors, but an accuracy requirement of 0.01 per cent. For this accuracy, $A(j\omega)$ has 90-deg phase shift at 100 cps. Under these conditions, $1 + [A(j\omega)/5]$ will be the sum of two vectors 90 deg out of phase. A cut-and-try method of establishing $A(j\omega)$ is required. Assume that $A(j\omega)|_{100\text{cps}}$ will be 10^4, or 80 db.

The error term then is $(2 \times 10^3/\underline{90}\ \text{deg})/(1 + 2 \times 10^3/\underline{90}\ \text{deg})$. The magnitude of $1 + 2 \times 10^3/\underline{90}$ deg is 2,000.00025. The error is $(2.5 \times 10^{-4})/(2 \times 10^3)$, or much less than 0.01 per cent.

Thus a gain of 80 db is more than satisfactory. A low-frequency gain somewhat greater than 80 db, with a corner frequency below 100 cps to keep the phase angle close to 90 deg at 100 cps, will give satisfactory amplitude accuracy. Note that, for this response, the phase error at 100 cps will be about 0.025 deg, which is greater than it would be if the phase angle of $A(j\omega)$ at 100 cps were 0 instead of 90 deg. This phase error can be tolerated, however, because of the phase characteristics of the computing resistors. For 0.01 per cent accuracy, wire-wound resistors must be used for computing. The resistors have a small

parallel capacitance, and this capacitance will vary at least ± 1 $\mu\mu$f for 1-megohm resistors; the phase shift of Z_f/Z_i, the multiplying expression in front of the error term, will be about 0.07 deg at 100 cps. The phase shift of the error term will, thus be small compared with the phase shift of Z_f/Z_i. It is possible to add capacitance across summing-amplifier resistors to equalize them (Sec. 2.4), but this is expensive in general-purpose computers.

Usually, the gain determined by the above method will not be sufficient for accurate integration at low frequencies. However, accurate integration requires automatic balancing circuits (Art. 2.3.14) which add to the low-frequency gain.

After the required amount of gain is determined, d-c amplifier stages can be designed, starting with the output stage.

DESIGN OF D-C AMPLIFIER STAGES

2.3.3. Choice of Output-stage Circuit. The output stage must be capable of delivering the required current and voltage ranges. It should also be designed to have low output impedance to minimize load-capacity effects on closed-loop amplifier stability. High-gain amplifiers can stand more output-stage distortion than low-gain amplifiers, since the distortion effect is reduced by the gain of preceding stages. Amplifiers should be capable of delivering rated voltage and current at all frequencies below 100 cps. Figure 2.3.3 illustrates different output circuits.

In general, low- or medium-μ tubes are required for output stages to have the required current and plate-dissipation capabilities. The choice of supply voltages involves a compromise between dynamic range and plate-dissipation limits; supply voltages of ± 250 to ± 300 volts are typical.

2.3.4. Voltage-amplifier Stages. Voltage-amplifier stages need not supply much current and may be designed with the aid of linear equivalent circuits based on published vacuum-tube parameters (Fig. 2.3.4). Symbols are defined as follows:

E_{bb}: supply voltage in the plate-cathode circuit
E_{cc}: supply voltage in the grid-cathode circuit
I_{b0}: quiescent (no signal) current into the plate
E_{b0}: quiescent plate-cathode voltage
E_{c0}: quiescent grid-cathode voltage
I_b: actual plate current with signal applied
E_b: actual plate-cathode voltage with signal applied
E_c: actual grid-cathode voltage with signal applied
$i_b = I_{b0} - I_b$: variation in plate current with signal applied
$e_b = E_{b0} - E_b$: variation in plate-cathode voltage with signal applied
$e_c = E_{c0} - E_c$: variation in grid-cathode voltage with signal applied
e_k = variation in cathode voltage with signal applied
$r_p = e_b/i_b$, when e_c is constant: dynamic plate resistance
$g_m = i_b/e_c$, when e_b is constant: transconductance
$\mu = e_b/e_c = r_p g_m$, when i_b is constant: amplification factor

V-C, or small-signal, linear equivalent circuits involve only small-signal vacuum-tube parameters. The tube manufacturer's data sheets give μ, r_p, and g_m for various operating points (Art. 2.3.5) In general, μ remains fairly constant over a wide operating range, but r_p decreases and g_m increases with increasing plate current.

The grid-plate capacitance C_{gp}, the grid-cathode capacitance C_{gk}, and the plate-cathode capacitance C_{pk} can affect high-frequency response and amplifier stability. Figure 2.3.5 shows the small-signal equivalent circuit of a triode stage. For pentode tubes, one may have to consider the dynamic effects of the screen unless the screen can be suitably by-passed.

Figure 2.3.6 shows a regenerative (positive-feedback) stage used to obtain large d-c gain from one tube.

2.3.5. Selection of Operating Points. Basic stage-design techniques utilize the published vacuum-tube characteristic curves. The quiescent plate voltage E_{b0} should be about half the plate supply voltage E_{bb} for stages near the amplifier output.

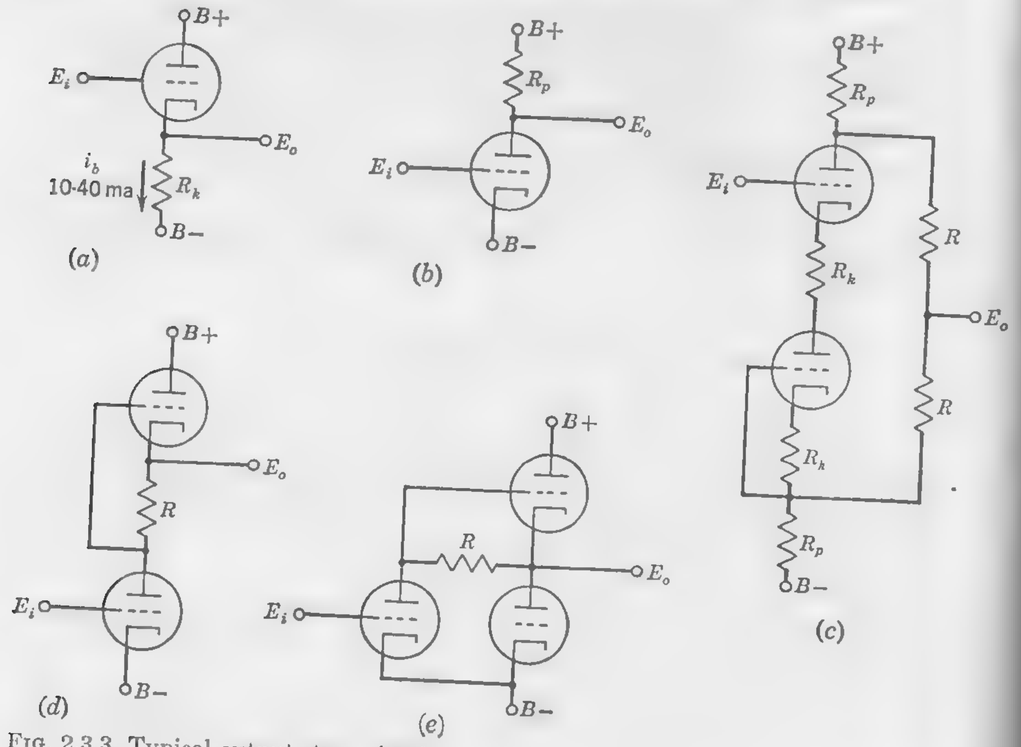


FIG. 2.3.3. Typical output-stage circuits.

FIG. 2.3.3*a*. Cathode follower. This circuit has low output impedance and less than unity gain. A disadvantage is the relatively high quiescent current required. The quiescent current must be greater than the maximum expected negative load current.

$$Z_0 \approx \frac{1}{g_m}$$

FIG. 2.3.3*b*. Plate-loaded amplifier. This circuit has relatively high output impedance. The quiescent current must be greater than the maximum expected positive load current. The gain is given by

$$A(s) = -\frac{\mu R_l}{r_p + R_l}$$

The output impedance is the parallel combination of r_p, R_l, and the load resistance.

FIG. 2.3.3*c*. Bridge-balanced amplifier (see also Ref. 2, p. 207).

FIG. 2.3.3*d*. Series amplifier (also referred to as "totem-pole" or "cascode" amplifier). This circuit has relatively low output impedance and requires little quiescent current. The no-load gain and the output impedance are given by

$$A(s) = \frac{E_o}{E_{in}} = \frac{-\mu(r_p + \mu R)}{2r_p + (1 + \mu)R}$$

$$Z_o = r_p\left(\frac{r_p + R}{2r_p + (\mu + 1)R}\right)$$

(Ref. 3, pp. 456–457).

FIG. 2.3.3*e*. Series amplifier. This arrangement also has low output impedance and low quiescent current but requires three tubes. The d-c gain obtainable with low-μ tubes is approximately 12 to 15. The output impedance is about 300 ohms.

A somewhat lower E_{b0} can be used for early stages which do not require a large dynamic range.

I_{b0} should be selected so that $g_m \geq 600$ micromhos. Even higher values of g_m are desirable for good high-frequency characteristics. I_{b0} must not be too large, since this requires either a small R_p or a large E_{bb}. R_p should be large compared with r_p, for

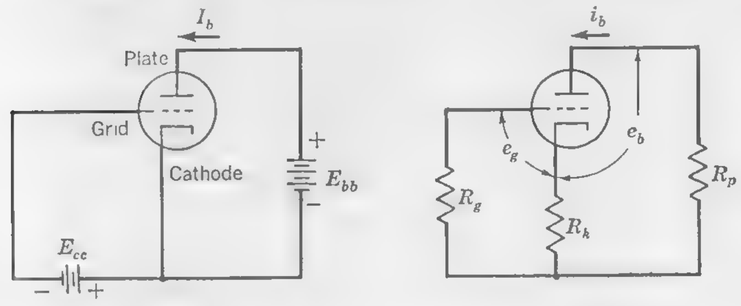


FIG. 2.3.4*a*. Triode amplifier.

FIG. 2.3.4*b*. Voltage designations of triode amplifier.

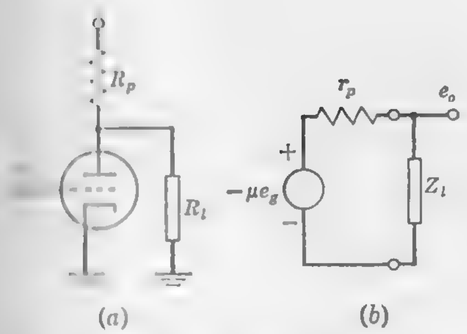


FIG. 2.3.5.

FIG. 2.3.5*a*. Triode amplifier with load impedance. (*b*) Triode amplifier small-signal voltage equivalent circuit. Gain is given by

$$A(s) = -\frac{\mu Z_i'}{r_p + Z_i'}$$

where Z_i' is the parallel combination of R_p, C_{pk}, and the input impedance of the coupling network to the following stage.

FIG. 2.3.6. Regenerative amplifier stage. The no-load stage gain is

$$A(s) = \frac{\mu Z_i'/[r_p + Z_i' + (1 + \mu)R_k]}{1 - \mu Z_i'\beta/[r_p + Z_i' + (1 + \mu)R_k]} \qquad \beta = \frac{Z_{ok}}{R_1 + Z_{ok}}$$

where Z_{ok} is the impedance working into the cathode of the first stage:

$$Z_{ok} = \frac{(Z_i' + r_p)R_k}{Z_i' + r_p + (1 + \mu)R_k}$$

The high stage gain obtainable can reduce the number of amplifier stages and hence also the number of transfer-function poles. Stages incorporating appreciable regeneration may require adjustments for gain and d-c level.

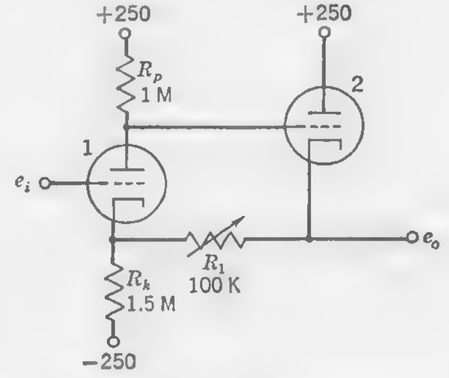


FIG. 2.3.6.

maximum low-frequency gain. The value of I_{b0} will affect the total plate power-supply current; a value of I_{b0} giving g_m of 1,000 to 1,500 micromhos is usually adequate. R_p can be determined from

$$R_p = \frac{E_{bb} - E_{b0}}{I_{b0}} \tag{2.3.7}$$

where E_{bb} is usually between 250 and 300 volts.

2.3.6. Effects of Cathode-bias Resistors and Screen Series Resistors.[2] If the grid bias for the operation of an amplifier tube is obtained by means of a cathode

resistor in the manner shown in Fig. 2.3.7*a*, the net voltage between grid and cathode becomes

$$e_g = e_i - i_p R_k \tag{2.3.8}$$

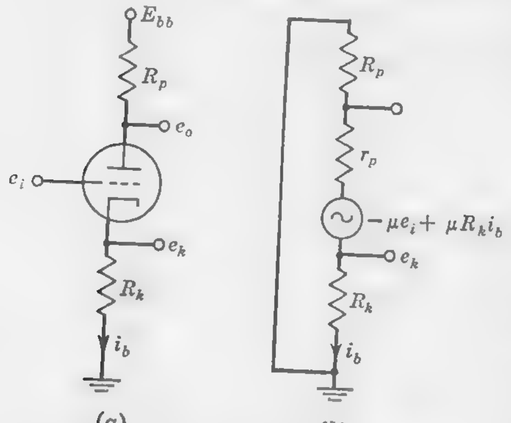
The current and voltage changes may again be derived from the equivalent linear circuit as shown in Fig. 2.3.7*b*. The grid-to-plate gain of the stage is

$$\frac{e_o}{e_i} = \frac{-\mu R_p}{R_p + r_p + (1+\mu)R_k} \tag{2.3.9}$$

where e_i is the input voltage, and e_o is the output voltage. The grid-to-cathode gain is found to be

$$\frac{e_k}{e_i} = \frac{\mu R_k}{R_p + r_p + (1+\mu)R_k} \tag{2.3.10}$$

where e_k is the cathode voltage change. All voltages are measured with respect to the point shown grounded in Fig. 2.3.7.



(*a*) (*b*)

FIG. 2.3.7. Triode with cathode degeneration and its equivalent circuit.

The values of r_p, g_m, and μ to be used in Eqs. (9) and (10) are again determined by the effective operating point. The latter may be found from the tube characteristics through successive approximation of the plate current i_b. A comparison of Eq. (10) and the equation given in Fig. 2.3.5 shows that a tube in the circuit of Fig. 2.3.7*a* acts like a tube of "equivalent dynamic plate resistance"

$$r_p' = r_p + (1+\mu)R_k \tag{2.3.11}$$

connected in the simpler circuit of Fig. 2.3.5. It is also seen that the inverse feedback due to the cathode resistor has decreased the gain of the stage. A similar effect is produced by a screen dropping resistor used with a pentode; the corresponding analysis is quite complicated. Because of the loss in gain, the use of cathode and screen dropping resistors to obtain correct electrode voltages is often avoided in favor of fixed-voltage sources. On the other hand, the degenerative effect, especially of the cathode dropping resistor, tends to reduce the effects of amplifier-gain changes and distortion.

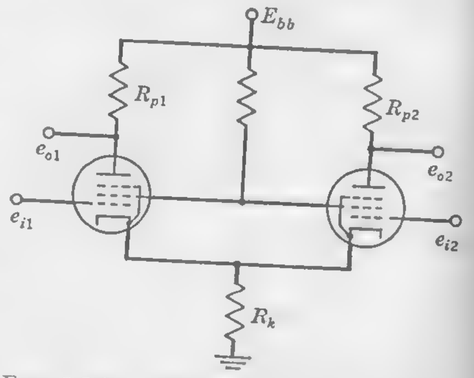


FIG. 2.3.8. Differential amplifier circuit. R_{p1} may be zero or different from R_{p2}. In addition, R_k may be returned to a negative voltage instead of ground. The tubes can be either pentodes or triodes.

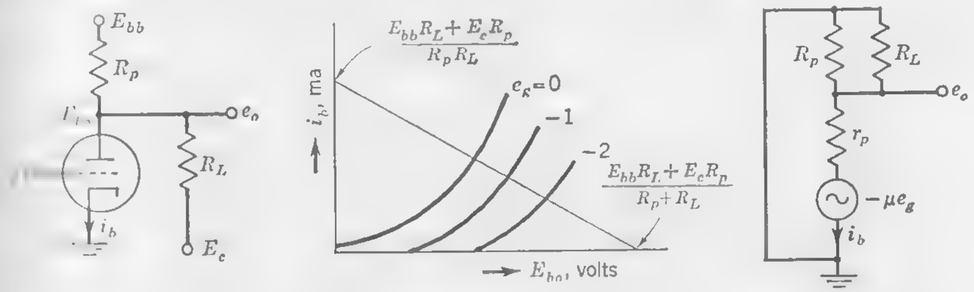
In differential amplifier circuits like the one shown in Fig. 2.3.8, the degenerative effect of the cathode and screen resistors is canceled, or at least reduced, because the plate-current changes in the two tubes oppose each other. In such circuits, approximately one-half the full-stage gain given in Fig. 2.3.5 can be realized in spite of the cathode and/or screen dropping resistors.

2.3.7. The Effect of Load Resistance.[2] Each computer amplifier stage is, in general, connected to a resistive load of resistance R_L, which may be returned to ground or to some fixed-voltage source of voltage E_c with respect to ground (Fig. 2.3.9*a*). One can analyze the effect of the load on the amplifier performance by drawing a new load line across the characteristics of the amplifier tube. Such a load line is shown in Fig. 2.3.9*b* for a triode amplifier and expresses the relation between the plate voltage e_b

and the plate current i_b as determined by the values of R_p, R_L, E_{bb}, and E_c. The value of i_b for many voltage-amplifier stages is so large that the effect of the load is negligible. When the operating point on some desirable load line has been chosen, it is possible again to use the linear equivalent circuit for the vacuum tube to derive the voltage and current changes. Figure 2.3.9*c* shows the linear equivalent circuit for Fig. 2.3.9*a*. The stage gain is now easily derived by substituting the total load resistance

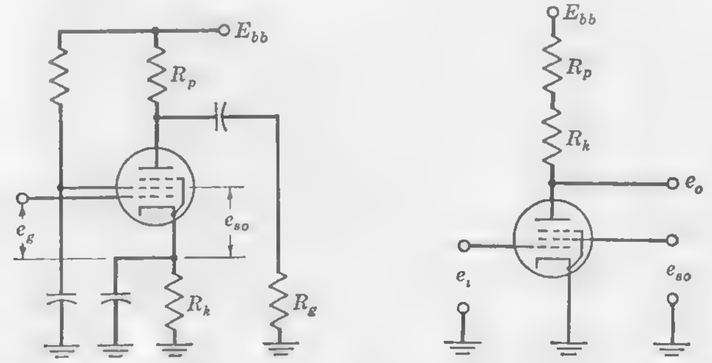
$$R_p' = \frac{R_p R_L}{R_p + R_L} \tag{2.3.12}$$

for the plate load resistance in the equation given in Fig. 2.3.5. Similar equivalent circuits may be used to derive the gain of other loaded amplifier stages.



(*a*) (*b*) (*c*)

FIG. 2.3.9. Triode amplifier with load resistance (*a*), load line on i_b, e_b characteristics (*b*), and equivalent circuit (*c*).



(*a*) (*b*)

FIG. 2.3.10. Development of d-c amplifier stage from a resistance-coupled a-c amplifier. Either triodes or pentodes may be used.

2.3.8. Use of Resistance-coupled-amplifier Tables.[2] To save time, it may be expedient to adopt vacuum-tube operating conditions similar to those given in standard resistance-coupled-amplifier tables. Such tables usually refer to an a-c voltage amplifier stage of the type shown in Fig. 2.3.10*a*, having a by-passed cathode resistor R_k. The tabulated data may be used to design an "equivalent" d-c amplifier, shown in Fig. 2.3.10*b*. This d-c amplifier will have approximately the same operating point as the original a-c amplifier if the resistance of the load connected to it is high; this is usually the case for voltage-amplifier stages. The equivalent amplifier has fixed E_{bb} and, in the case of pentodes, a fixed screen voltage. The gain will, in general, be slightly higher than that given in the tables, since the plate load resistance is higher and also because the resistance of the load connected to the d-c amplifier stage often will be higher than the resistance R_g specified in the tables.

In a-c amplifiers, the value of the plate load resistance R_p is often limited by the high-frequency response required of the stage. In d-c amplifiers, it is often permissible to use rather large values of R_p and correspondingly small plate currents in order to obtain a large voltage gain. In the case of pentodes, maximum stage gain

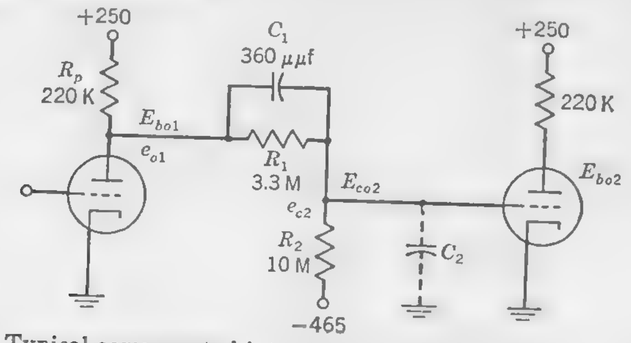


FIG. 2.3.11. Typical compensated interstage coupling network ($E = -465$ volts).

can be obtained by using low screen voltage as well as low plate voltages. The lowest permissible values of plate and screen voltages will be determined by the amount of positive grid current which can be tolerated.

2.3.9. Interstage Coupling Networks. Figure 2.3.11 shows a typical interstage coupling network which provides conductive coupling (d-c operation) at the correct operating voltage E_{c02} for the second grid. R_1 and R_2 should be large compared with the plate resistance r_p and the plate load resistance R_p of the first tube. One computes the required value of the ratio R_2/R_1 from

$$\frac{R_2}{R_1} = \frac{E_{c02} + E}{E_{b01} - E_{c02}} \tag{2.3.13}$$

Values of R_1 and R_2 can then be selected so that $(R_1 + R_2)$ is much larger than R_p.

C_1 is shunted across R_1 to compensate partially for the effective input capacity of the second stage. The input impedance of a vacuum-tube-amplifier stage is determined from the circuit of Fig. 2.3.12:

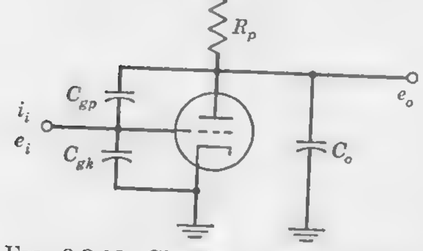


FIG. 2.3.12. Circuit for determining input impedance of vacuum tube.

$$Z_{in} = \frac{e_{in}}{i_{in}} \approx \frac{1}{[1 + A(s)]sC_{gp} + sC_{gk}}$$

where $A(s)$ is the grid-to-plate gain of the amplifier stage and is dependent on C_o and the input impedance of the following stage.

$$Z_{in} = \frac{s + \alpha}{s(C_{gp} + C_{gk})(s + \alpha) + K_1\alpha C_{gp}s} \tag{2.3.14}$$

where

$$\frac{1}{\alpha} = \left(\frac{r_pR_p}{r_p + R_p}\right) C_o$$

K_1 = d-c gain of the second stage

The input impedance of the second stage, in parallel with R_2, gives a second-order response which cannot be completely compensated by a single capacitor. C_1 causes the transfer function of the interstage network to be of second order in both numerator and denominator, i.e.,

$$\frac{e_{c2}}{e_{o1}} = K_2\frac{s^2 + a_1s + a_2}{s^2 + b_1s + b_2} \tag{2.3.15}$$

In general, e_{c2}/e_{o1} will have two zeros and two poles. By properly selecting C_1, two pole-zero pairs are formed in which each pole-zero pair, or dipole, has sufficiently small separation, so that the interstage network will have negligible effect on the open-loop frequency response. This will be true if R_1C_1 is approximately $K_1C_{gp}R_2$; C_1 can be somewhat smaller to reduce the effective capacity shunting the first-stage output, if desired.

One can minimize the attenuation due to the interstage network by making E as large as possible. Also R_1 can be a device with large d-c resistance and small dynamic resistance (neon tubes or Zener diodes). For reasonable values of E, the attenuation due to resistive networks is not objectionable.

Resistance tolerances permissible for R_1 and R_2 depend on the location of the network in the amplifier. If e_{o1} requires almost the full dynamic range of the plate, R_1 and R_2 may have to be 1 per cent resistors. For early stages, 5 or 10 per cent resistors are generally sufficient. To be safe, the network coupling the output stage to its driver should consist of 1 per cent resistors of sufficient power rating.

FREQUENCY STABILIZATION: ROLL-OFF NETWORKS

2.3.10. High-frequency Stabilization. After the various stages and their coupling networks are designed, the open-loop frequency response must be shaped to yield stable closed-loop operation. As a starting point, a frequency-gain curve (Fig. 2.3.2)

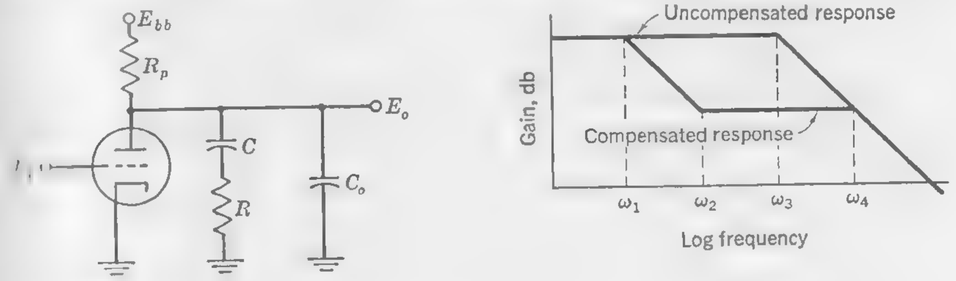


FIG. 2.3.13. Typical series RC compensated stage (a), and gain-frequency plot of compensated and uncompensated stage (b). For $r_p \ll R_1$, $r_p \ll R$, $C_o \ll C$,

$$\omega_1 \approx 1/r_pC$$
$$\omega_2 \approx 1/RC$$
$$\omega_3 \approx 1/r_pC_o$$
$$\omega_4 \approx 1/RC_o$$

may be selected. For amplifiers which are to be operated with a variety of input-feedback network combinations, it is usually desirable to have an open-loop response which goes through unity gain with a slope approximating 20 db per decade; special-purpose operational amplifiers must be designed with the particular application in mind.

The reason for going through unity gain on a 20 db per decade slope is to ensure that the amplifier has no more than 90-deg phase shift at unity gain. This, in turn, helps to ensure that the phase shift of the second term of Eq. (1) is less than 180 deg when its magnitude is unity. The Nyquist condition for stable operation is then satisfied. Roll-off networks are inserted to reduce the gain at the desired rate.

2.3.11. Roll-off or Lag-lead Networks. The most commonly used type of roll-off network is a series RC circuit from plate to ground (Fig. 2.3.13). If we assume that the coupling network to the following stage has a low-frequency impedance large compared with r_p and R_p, then its effect is primarily that of the input capacity to the coupling network, represented by C_o. Actually, because of the effect (Miller effect) discussed in Art. 2.3.9, C_o is not a pure capacitance. For most purposes, however, C_o is the series combination of the actual input capacity of the following stage and the compensating capacitor of the interstage network.

The transfer function of the stage with compensation is

$$\frac{e_o}{e_i} = \frac{-\mu R\left(s + \frac{1}{RC}\right)}{r_p\left[RC_o\left(s + \frac{1}{R_pC_o}\right)\left(s + \frac{1}{RC}\right) + s\right] + R\left(s + \frac{1}{RC}\right)} \tag{2.3.16}$$

Equation (16) is somewhat complicated but can be simplified by certain assumptions. If one assumes $R_p \gg r_p$, $R \ll r_p$, $C \gg C_o$, then Equation (16) can be approximated by

$$\frac{e_o}{e_{in}} = \frac{\mu R_p}{r_p + R_p}\left[\frac{\left(s + \frac{1}{RC}\right)}{r_p C_o \left(s + \frac{1}{r_p C}\right)\left(s + \frac{1}{RC_o}\right)}\right] \tag{2.3.17}$$

Equation (17) shows that the d-c gain is unchanged, a zero is produced at the break point of RC, a pole at the break point of r_pC, and a second pole at the break point of RC_o. From Fig. 2.3.13, $\omega_1 = 1/r_pC$, $\omega_2 = 1/RC$, and $\omega_4 = 1/r_pC_o$. The value of C determines the frequency at which the roll-off starts, and the ratio R/r_p determines the attenuation produced. Note that the zero at $\omega_2 = 1/RC$ is exact; it does not depend on the approximations involved.

The uncompensated stage has a pole at ω_3 which will be at a frequency below 1 Mc. The corresponding high-frequency pole of the compensated stage is moved out a distance very nearly equal to the attenuation produced, i.e., ω_3/ω_4 is approximately equal to R/r_p. Thus uncompensated high-frequency poles can be moved out to frequencies at which their effect on stability is greatly reduced. By frequency-compensating several stages one can cause the open-loop frequency response to have the desired 20 db per decade slope out to beyond 1 Mc.

In a multistage amplifier it is usually desirable to frequency-compensate all stages except the output stage and possibly one other stage. If each compensated stage has an attenuation approximately equal to its d-c gain, then the residual high-frequency break points, corresponding to ω_4, will all be moved out to the region beyond 1 Mc, provided that the gain-bandwidth product of each stage is greater than 1 Mc. If one stage takes all the attenuation, the remaining stages will have their ω_3 break points below 1 Mc. Because the external circuitry produces poles at relatively low frequencies, it is desirable to have no more than one amplifier pole below 1 Mc

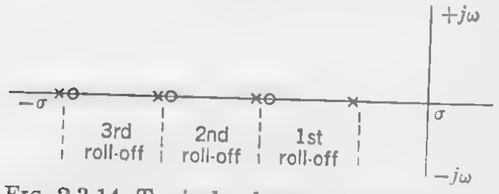


FIG. 2.3.14. Typical pole-zero plot of multistage amplifier with compensation in several stages.

The roll-offs of the various stages should be spaced so that the total roll-off is reasonably smooth. Assume the lowest-frequency roll-off network has its pole at the frequency determined from Fig. 2.3.2, and a zero such that the ratio between the pole and zero frequencies is equal to the low-frequency gain. The next roll-off network then would have its pole corresponding to the zero of the first network. If one stage is to be left uncompensated, the last zero produced by a roll-off network should be at the pole position of the uncompensated stage. Figure 2.3.14 illustrates a pole-zero plot of an amplifier with several roll-off networks.

Exact pole-zero cancellation is not necessary. Twenty per cent components can generally be used, since slight miscancellation has a negligible effect on the over-all closed-loop response.

Certain other considerations enter into the determination of where the roll-off networks will be placed in the amplifier. The roll-off network must not cause any stage to draw excessive current at any desired signal frequency; so the maximum signal levels at each stage must be considered. This may make it impossible to take the maximum attenuation desired in all stages.

Another effect to be considered is that of operation of stages preceding the roll-off. As the gain of a particular stage is reduced, the stages ahead must have larger signal swings in order to produce the desired output amplitude. This effect may limit the frequency at which maximum output amplitude can be obtained. If certain stages are required to have large swings, the amplifier distortion may increase.

It is difficult to specify rules which will work in all cases. In general, though, the

[illegible]-frequency roll-offs are taken in the earlier stages of the amplifier. Considerable shuffling may be required to meet maximum amplitude and distortion requirements, or these requirements may have to be determined on the basis of what can be accomplished. Several compromises may have to be established. That is, the specifications on frequency to which maximum amplitude can be obtained at less than a certain distortion are all interdependent with stability considerations. Distortion must be kept sufficiently low for the desired computing accuracy. For example, an amplifier designed to have less than 0.01 per cent error must have less distortion than [illegible] at the maximum desired computing frequency (usually 100 cps).

2.3.12. External Compensating Networks. Compensation can also be accomplished by the external circuitry associated with the amplifier. Addition of capacity across

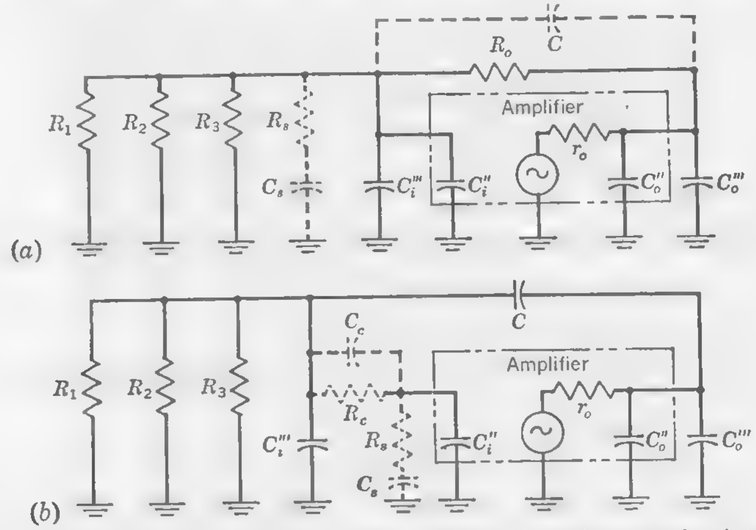


FIG. 2.3.15. Equivalent circuits for a summing amplifier (*a*) and a summing integrator (*b*). [illegible] the amplifier source impedance, and C_i'' and C_o'' are amplifier input and output [illegible] capacitances. C_i'', C_i''', and C_o''' include the effects of cables and patchbays, which may give rise to shunt capacitances as large as 1,500 $\mu\mu$f. Possible stabilization networks are indicated in broken lines; sometimes C_i'' is increased artificially to improve stability. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

summer feedback resistors will tend to stabilize (Sec. 2.4). Capacity added from summing junction to ground will help stabilize amplifiers with very low feedback resistance or large capacitive feedback (e.g., integrators; see also Fig. 2.3.15).

DRIFT COMPENSATION AND AUTOMATIC BALANCING CIRCUITS

2.3.13. Offset Voltages and Drift. For balancing of computer amplifiers, the output d-c reference level of the d-c amplifiers used as analog-computing elements must be equal to the input (grid) d-c reference level (usually zero with respect to ground) in order to permit d-c feedback and interconnections of computing elements. It is necessary to have some means of adjusting the output to zero for zero input. Such an adjustment is termed the balancing or zero adjustment. Figures 2.3.16 and 2.3.17 show typical arrangements for balancing summer and integrator amplifiers. See also Ref. 2, pp. 191ff., for a good discussion of drift-reduction methods.

2.3.14. Chopper Stabilization. Figure 2.3.18 illustrates a typical open-loop block diagram of a chopper-stabilized d-c amplifier. Normally, $A_1(s)$ is simply unity, but other configurations may be used (Fig. 2.3.19).

An input capacitor, as in Fig. 2.3.19*b*, may be used to minimize the effect of input grid current. The capacitor should be a good-quality polystyrene for maximum leakage resistance and minimum soakage effect. Soakage is the tendency of a capaci-

tor to maintain a small d-c voltage after being charged. The time for this charge to disappear completely, even with a low-resistance discharge path, varies with the type of capacitor. Paper capacitors are poor; Mylar dielectric capacitors are considerably better, and polystyrene capacitors are better than Mylar.

A disadvantage of this circuit is that the capacitor charges during overload periods and may require a relatively long time to discharge. It is possible to shunt the

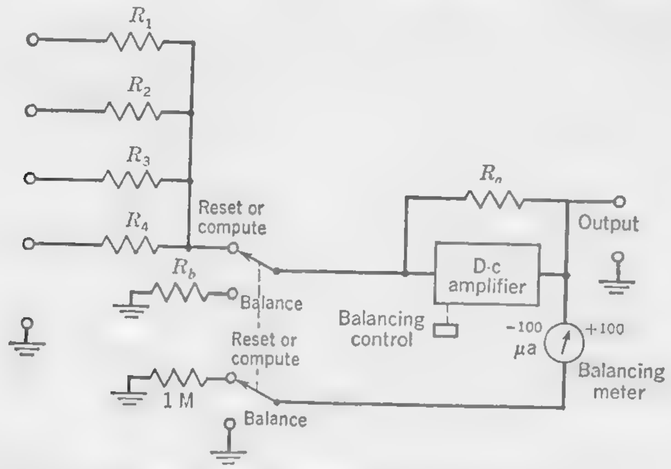


FIG. 2.3.16. Switching arrangement for balancing a summing amplifier. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

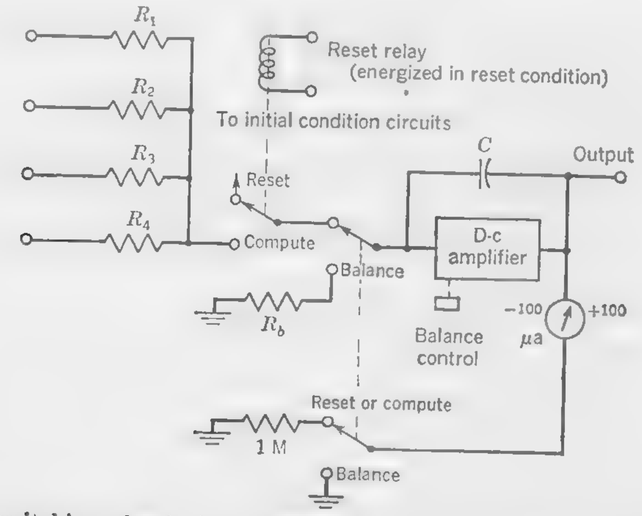


FIG. 2.3.17. A switching circuit for balancing a summing integrator. The integrating capacitor C is often replaced by a feedback resistor for balancing. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

capacitor with relay contacts which are closed in RESET; alternatively, one may connect Zener diodes between summing point and ground.

The circuit of Fig. 2.3.19*c* helps to filter out harmonics generated by the chopper action without an excessively large time-constant filter at the stabilizer output. This reduces overload recovery time and still permits adequate filtering. The stabilizer harmonics introduced ahead of $A_3(s)$ at the mixing point are reduced by the gain ahead of that point.

The circuit of Fig. 2.3.19*d* is a combination of the second and third types

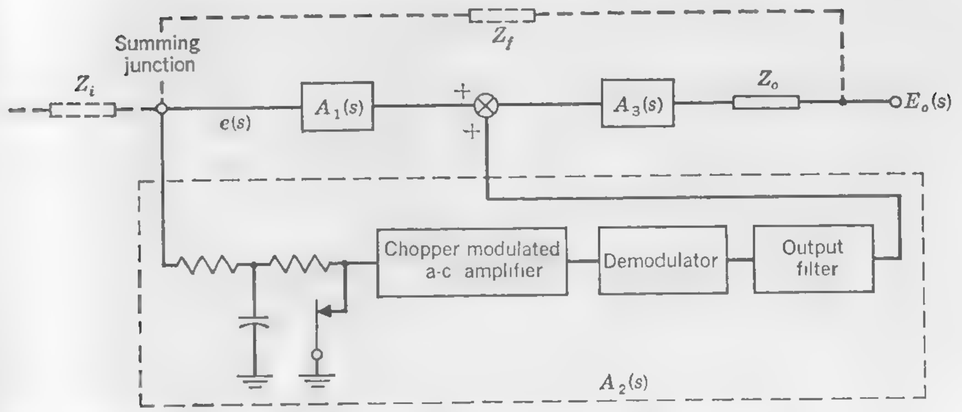
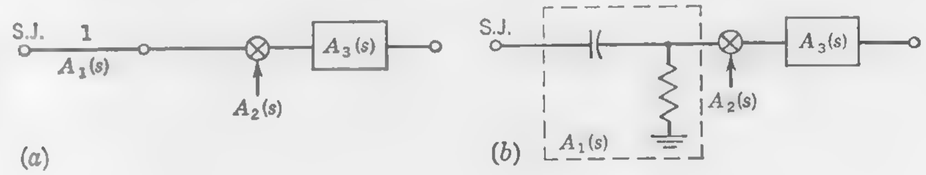


FIG. 2.3.18. Block diagram of a typical chopper-stabilized d-c amplifier.

$$\frac{E_o(s)}{e(s)} = A_3(s)[A_1(s) + A_2(s)]$$



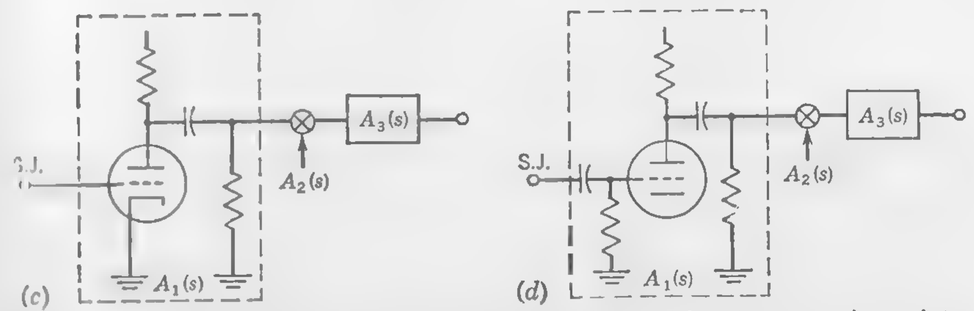


FIG. 2.3.19*a*. Block diagram of stabilized amplifier with direct input to summing point.
FIG. 2.3.19*b*. Block diagram of stabilized amplifier with capacitor input to summing point.
FIG. 2.3.19*c*. Block diagram of stabilized amplifier with capacitively coupled single stage of gain ahead of summing point.
FIG. 2.3.19*d*. Block diagram of stabilized amplifier with capacitor input and capacitively coupled stage of gain ahead of summing point (summing point mixes the stabilizer signal and high-frequency signals).

2.3.15. Design of Chopper-stabilizer Circuits. The over-all open-loop transfer function of the chopper-stabilized amplifier is

$$\frac{E_o}{e(s)} = A_3(s)[A_1(s) + A_2(s)] \tag{2.3.18}$$

$A_1(s)$ and $A_2(s)$ must be selected so that the magnitude of their sum does not decrease to a value too low for accurate computation, i.e., their individual frequency responses must be carefully matched. It is possible to analyze this sum on a frequency-response basis by vector addition of $A_1(j\omega)$ and $A_2(j\omega)$ through the range of frequencies over which they operate. This is a tedious process: the stabilizer transfer function $A_2(s)$ is not a simple ratio of two polynomials in S due to the action of the chopper. Sampled-

data techniques can be used for this analysis, but they are involved, and synthesis of the proper networks is even more difficult. Fortunately, $A_2(s)$ is dominated by its input and output filters, and a reasonable approximation can be made by assuming the gain between the filters to be independent of frequency.

Figure 2.3.20 shows a typical chopper-stabilizer circuit using a single chopper for modulation and demodulation. Separate choppers may be used, especially when amplifiers are built in groups of two or four; more frequently, the output chopper (which is less sensitive to drift) is replaced by a diode switching circuit (Figs. 2.5.4 and 2.5.6).

In Fig. 2.3.20, K_2 is an a-c–coupled amplifier. Its gain to the frequencies of the chopper modulator and its first few harmonics is determined by the amount of drift reduction desired. Increasing K_2 lowers the amplifier drift rate and decreases the frequency of zero adjustment required (automatic balancing circuits still require a

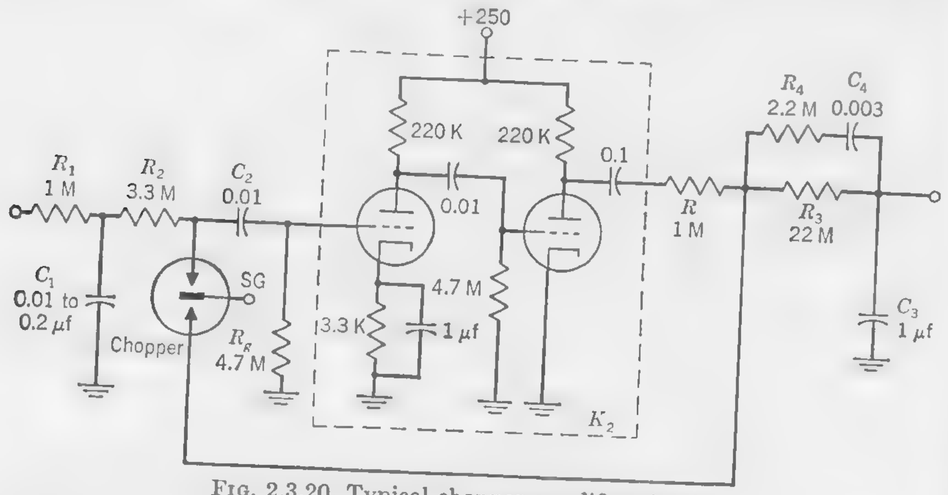


FIG. 2.3.20. Typical chopper amplifier circuit.

"fine" zero adjustment for optimum accuracy). The sign of $A_2(s)$ (inverting or noninverting) depends on $A_3(s)$. $A_2(s)$ must be inverting if $A_3(s)$ does not invert, which is usually the case. For the single chopper circuit shown, K_2 must be noninverting for $A_2(s)$ to invert, which means K_2 must be composed of an even number of stages. When separate choppers are used, the choppers can be phased to give the proper polarity for $A_2(s)$, in which case K_2 can have either an even or an odd number of stages. When K_2 has an even number of stages, and one chopper is used, the chopper must have make-before-break contacts to prevent oscillations while the swinger is moving from one contact to the other.

R_1 and R_2 are generally in the range of 1 to 3 megohms, with R_2 being larger than R_1. R_g should be larger than $(R_1 + R_2)$ to prevent excessive attenuation to the error signal. The input filter is not a true RC filter because of the periodic switching action; but for $R_2 > R_1$, the filter can be assumed to have a time constant R_1C_1. This approximation is not rigorous, but it simplifies the analysis without introducing serious error.

R_1C_1 serves two purposes: it attenuates pickup of the chopper-driving frequency in the input circuitry, and it attenuates high signal frequencies at the summing junction so that they are not amplified in the stabilizer amplifier. The summing-junction signal increases at the rate of 20 db per decade because the open-loop gain decreases at that rate for constant output amplitude. If the amplifier is expected to maintain full output amplitude to 1,000 cps or more, then the summing-junction signal will be fairly large at these frequencies. During the time the modulator contacts are open, any summing-junction signal appearing at the grid of the first stabilizer tube will be amplified. This signal must be kept small to prevent swamping the stabilizer amplifier. To determine R_1C_1, the maximum summing-junction signal must be calculated,

the amount of signal the stabilizer can handle without affecting its action must be determined, and R_1C_1 must be selected to give the required attenuation. R_1C_1 varies from 0.01 sec for low stabilizer gains to 0.22 sec for high gains (10,000 to 20,000). R_1C_1 should be no larger than necessary for stability.

C_1 should be a Mylar or polystyrene capacitor for minimum soakage effect. C_1 will charge to a large voltage on overload. The amplifier will come back to balance with an offset, determined by the soakage potential, which may be several hundred microvolts. It will require several minutes to stabilize fully unless a good capacitor is used.

Note that the grid resistor of the first stabilizer stage is grounded, and a by-passed cathode resistor is used to establish bias. If fixed bias were used, a portion of the bias voltage would be chopped because of the leakage resistance of C_2, which forms a voltage divider together with R_g. This leakage resistance varies with temperature and would cause drift which is not compensated by the stabilizer gain. Grounding R_g eliminates this effect. In addition, the use of Mylar or polystyrene for C_2 will minimize the temperature effect.

The output filter must eliminate the harmonics generated by the chopper action. These frequencies are very low, and R_3C_3 should have a time constant somewhat greater than 20 sec. For low stabilizer gains R_3C_3/R_1C_1 should be approximately equal to K_2. This is difficult to do with practical values of R_3C_3 if R_1C_1 is large. Also, increasing R_3C_3 causes a large increase in overload recovery time. C_3 charges up on overload and requires a long time to discharge after the overload is removed. R_3C_3 can be smaller than indicated by the previous criterion if R_4C_4 is added. R_4 should be small compared with R_3, and C_4 should be small compared with C_3. Under these conditions, the output filter will have poles determined by R_3C_3 and R_4C_4 with a zero determined by R_3C_4.

Returning now to the expression

$$A_1(s) + A_2(s) = A_1(s)\left(1 + \frac{A_2(s)}{A_1(s)}\right) \tag{2.3.19}$$

$$A_2(s) \approx \frac{K_2}{4}\left[\frac{\left(s + \frac{1}{R_3C_4}\right)}{R_1R_4C_1C_3\left(s + \frac{1}{R_1C_1}\right)\left(s + \frac{1}{R_3C_3}\right)\left(s + \frac{1}{R_4C_4}\right)}\right] \tag{2.3.20}$$

Equation (19) will have two complex zeros, plus real poles and zeros. The filter parameters should be selected so that the complex zeros are at less than a 45-deg angle from the negative real axis of the s plane. This will ensure proper matching of $A_1(s)$ and $A_2(s)$ to prevent the magnitude of their sum from becoming too small at any frequency. Application of root-locus techniques to $1 + A_1(s)/A_2(s)$ helps to determine the values of the various R's and C's.

An alternate experimental approach can be used to find R_4 and C_4 after R_1, C_1, R_3, and C_3 are selected. Connect the amplifier as a unity inverter with the input resistor grounded, and observe the output on an oscilloscope. Temporarily short the junction of R_3 and R_4 to ground, until the output has moved slightly off zero. Remove the short and observe the output return to zero with a response similar to an underdamped second-order system. Adjust R_4 and C_4 until the response exhibits less than one or two cycles of ringing. Be careful not to let the output move so far from zero that the error signal saturates the stabilizer amplifier.

2.3.16. Commutator Stabilization.[2] The alternative automatic balancing scheme illustrated by Fig. 2.3.21 utilizes a single stabilizing amplifier and a commutator-type switch to balance as many as 40 separate d-c amplifiers in periodic succession. The error voltage (summing-point voltage) E of the operational amplifier shown in Fig. 2.3.21*a* is sampled by a commutator about three times per second. Whenever the error voltage E differs from zero, the commutator forms a pulse which is amplified by the pulse amplifier shown. Another section of the commutator returns the amplified pulses to a ripple filter whose d-c output voltage is applied to the balancing grid of the d-c amplifier. Other pairs of commutator sections serve other d-c ampli-

fiers, so that the stabilizing-pulse amplifier is utilized on a time-sharing basis. The action of this automatic balancing circuit on each operational amplifier served is in principle similar to that of individual automatic balancing circuits based on chopper stabilization; the d-c gain of the circuit is again given by Eq. (18).

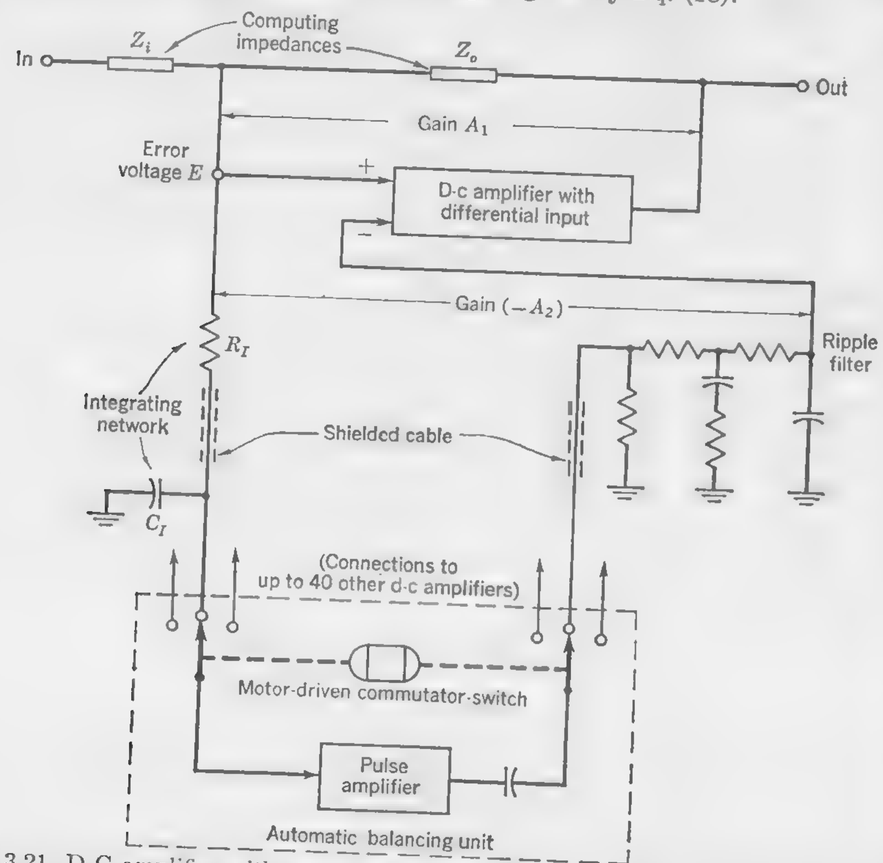


FIG. 2.3.21. D-C amplifier with commutator stabilization. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

MISCELLANEOUS DESIGN CONSIDERATIONS

2.3.17. Low Grid-current Input Stages. Input grid current to the amplifier causes offset errors which are especially serious in integrators.

Negative grid current is minimized by operating the first grid at least 1 volt negative with respect to the cathode. Positive grid current is due to the flow of positive ions to the grid. These positive ions are caused by collisions between the cathode-current electrons and the residual gas molecules in the tube. Electrometer tubes are designed to have grid currents of less than 10^{-14} amp, but they are expensive and provide very little gain. Ordinary low-μ tubes can be operated at quiescent plate voltages of about 20 to 25 volts and quiescent plate currents of about 50 μa to reduce ionization. This method will give a grid current of less than 10^{-10} amp.[6] Cathode-follower input circuits are also useful (Fig. 2.5.1).

Grid-blocking capacitors are also used in amplifiers which have automatic balancing circuits. High-quality polystyrene capacitors can be used. Their disadvantage is that they acquire a large charge when the amplifier is overloaded or saturated. When this happens, a relatively long time is required for the capacitor to discharge. The time can be shortened by connecting a relay to discharge the capacitor after an overload, or to prevent its acquiring a charge.

References

1. Nyquist, H.: Regeneration Theory, *Bell System Tech. J.*, II (January, 1932). Bode, H. W.: *Network Analysis and Feedback Amplifier Design*, chap. 8, D. Van Nostrand Company, Inc., Princeton, N.J., 1945.
4. Korn, G. A., and T. M. Korn: *Electronic Analog Computers*, chap. 5, McGraw-Hill Book Company, Inc., New York, 1956.
5. Valley, G. E., and H. Wallman: *Vacuum Tube Amplifiers*, McGraw-Hill Book Company, Inc., New York, 1948.
6. Evans, W. R.: *Control-system Dynamics*, McGraw-Hill Book Company, Inc., New York, 1954.
7. Truxal, J. G.: *Automatic Feedback Control System Synthesis*, McGraw-Hill Book Company, Inc., New York, 1955.
8. Meneley, C. A., and C. D. Morrill: Linear Electronic Analog Computer Design, *Proc. Natl. Electronics Conf.*, **5**:58 (1949).
9. Truxal, J. G.: *Control Engineers' Handbook*, sec. 6, McGraw-Hill Book Company, Inc., New York, 1958.
10. Cockrell, W. D.: *Industrial Electronics Handbook*, sec. 46, McGraw-Hill Book Company, Inc., New York, 1958.

PART 4: LINEAR-ELEMENT ERROR SOURCES AND EFFECTS

By CHARLES H. SINGLE

INTRODUCTION

2.4.1. Scope. This section presents a discussion of linear-element error sources and their effect on problem-solution accuracy in an analog computer. Balanced design is stressed; that is, no error source receives undue emphasis with resultant degradation of over-all system performance.

2.4.2. Basic Considerations. *Static accuracy* is achieved through high-gain low-drift d-c amplifiers (Sec. 2.3), precise and stable resistive network elements, and stable coefficient-setting potentiometers of high resolution (Sec. 2.1). *Dynamic accuracy* and *stability margin* are of at least equal importance. For optimization of these a-c characteristics, the *composite* system transfer function should approach its ideal most closely. Contributing a-c factors include capacitor stability, electrical purity of network resistors and capacitors, potentiometer capacitive loading, and amplifier open-loop frequency response and output impedance. The interaction of these factors is further complicated by such practical considerations as patchbay wiring (Art. 4.4). Resultant optimum system performance is generally achieved with *non-optimum* individual elements, particularly if compared with the more ideal individual bench characteristics.

For many linear-element configurations, the typical system-stability margin will be sufficient. The principal error source will then be from net system phase shift, since even slight phase shifts can combine to produce significant dynamic errors (Art. 2.4.3).

Greater system-stability margin is of tremendous value for accurate simulation of complex and/or high gain loops of either differential or algebraic form (Art. 2.4.14). It is also advantageous with unusual feedback configurations (limiters, etc.) or with capacitive loading, such as occurs in driving remote equipment.

2.4.3. Dynamic Errors vs. Stability Considerations. Generalization of the error analyses in Secs. 2.2 and 2.3 shows that the transfer function of any combination of linear computing elements (such as a cascaded coefficient potentiometer and summing amplifier) can be written as

$$\frac{X_0(s)}{X_1(s)} = Y(s)[1 - e(s)] \qquad (2.4.1)$$

where $Y(s)$ is the *desired* transfer function, and $e(s)$ is a *fractional-error term* due to

stray capacities, finite amplifier gain, etc. For sinusoidal input and small errors at normal computing frequencies ($\omega/2\pi < 100$ cps), $e(j\omega)$ causes essentially only phase shift (negligible *amplitude error*). The *largest possible absolute value of the fractional error due to phase shift* $|e(j\omega)|_{max}$ is approximately equal to the spurious phase shift in radians, or

$$|e(j\omega)|_{max} \approx |\arg [1 - e(j\omega)]| \approx \omega \left| \sum_{k=1}^{n} \tau_k + \sum_{j=1}^{m} 2\zeta_j \tau_j \right| = \omega|\tau_0| \qquad (2.4.2)$$

where τ_0 is the *equivalent first-order time constant* formed by the *algebraic sum* of the inevitable high-frequency poles and zeros that occur in the practical system. The lead or zero terms are taken as positive, lag or pole terms are taken as negative, and ζ_j is the damping ratio of quadratic terms.

Thus, if the system's maximum fractional error due to phase shift is to be less than 0.2 per cent at 100 cps or 0.02 per cent at 10 cps, the equivalent first-order time constant τ_0 must be less than 3.5 μsec.* The break-point frequency ($\omega\tau_0 = 1$) is 50,000 cps for this case. This again is the desired net effect from *all* computing elements of the system in question (e.g., potentiometer and summing amplifier).

A conventional method of achieving low phase-shift error without requiring the rather high bandwidth cited above is to allow individual computing elements to have a quadratic characteristic with damping ratio less than unity. This method inevitably entails a reduced stability margin or a tendency to oscillate, particularly in problems involving complex loops or loops with high gain. Fortunately, techniques have been developed that not only achieve the desired small phase shift at computing frequencies without major amplifier redesign but also increase system-stability margin at the same time (Art. 2.4.14).

Errors for particular linear differential equations have been investigated in some detail though substitution of various nonideal transfer functions for the ideal linear elements.[1,2,3,4] Articles 2.4.11, 2.4.14, and 2.4.15 deal with improved, more realistic analyses of this type.

COMPUTING RESISTORS

2.4.4. Resistor Specifications and Matching (see also Sec. 4.7). Most computing resistors are wire-wound because of the well-established long-term stability characteristics of such resistors. This situation may well change with continuing development of other resistor types. Typical electrical specifications for wire-wound computing resistors include:

Value:	
Nominal, ohms	$(10)^5$–$(10)^6$
Relative, ppm	20–1,000
Absolute, ppm	50–1,000
Stability, ppm per year	100–1,000
Temperature coefficient, ppm per °C	5–25
Power rating, watts	¼–1
A-C characteristics (Art. 2.4.5)	Uncontrolled or controlled
Packaging	Nonencapsulated or encapsulated

Factors that make such specifications somewhat inadequate are subtle. The value specifications may hold only at much less than rated power because of self-heating (without very special temperature control). Thermal potentials may cause errors (even with only *slight* temperature gradients across the resistors). Encapsulation in plastic can aggravate self-heating and may not be needed except under high-humidity conditions. The best recommendation is to evaluate resistor samples carefully in the expected environment.

The method used to match computing resistors can limit the flexibility of the computer. It is possible to achieve excellent summing-amplifier d-c accuracy with many

* Note that this is a true percentage, not the per cent of full scale usually specified.

ent resistance values in the machine if resistors with the desired *precise ratios* are grouped with *specific* amplifiers. The more flexible arrangement, where a large portion of the total resistor complement is unassigned (that is, can be associated with any desired amplifier by patchboard wiring) requires more precise resistors for the same ratio accuracy. Precise ratios can be established conveniently with 10 *matched* 100 kilohm resistance standards. The popular values (100, 200, 250, 500 kilohms and 1 megohm) can be obtained through direct computer comparison or bridge measurement.[5]

The main portion of a computing resistor is, typically, a carefully aged and temperature-cycled wire-wound resistor trimmed with a selected or adjustable wire-wound resistor. Trimming should be done at the operating temperature.

Component ovens generally furnished with large and medium-sized analog computers (over 30 amplifiers) can virtually eliminate dependence on air-conditioning equipment for network elements. Units are available that permit maintenance of resistor values to within ±15 ppm for ambient temperature between 65 and 110°F.

2.4.5. A-C Characteristics of Wire-wound Resistors. The a-c characteristics of wire-wound resistors can be approximated by the simplified equivalent circuit of Fig. 2.4.1 for frequencies less than 100 kc. The transfer impedance of this circuit is

$$Z_t(s) = \frac{E_r}{i_r}(s) = R_{dc} \frac{\frac{L}{R_{dc}}(s+1)}{LCs^2 + R_{dc}Cs + 1} = \frac{1}{C} \frac{s + \frac{R_{dc}}{L}}{s^2 + (R_{dc}/L)s + (1/LC)} \qquad (2.4.3)$$

This expression constitutes a more correct transfer impedance than just R_{dc} for the operational-amplifier performance equation (Secs. 2.2 and 2.3).

For wire-wound resistors, L and C can be held within limits given by

$$0 < L < (10)^{-7} R_{dc} \qquad \frac{2(10)^{-6}}{R_{dc}} < C < \frac{4(10)^{-6}}{R_{dc}} \qquad (2.4.4)$$

(units in ohms, henrys, and farads). Limit values differ for other resistor types. For the smaller values of R_{dc} (less than 100 kilohms) parasitic *inductive* effects tend to predominate, whereas for larger R_{dc} (greater than 1 megohm) the parasitic *capacitance* is most important. For conventional noninductively wound resistors, the break-point frequency of the L_p/L zero will be beyond 1.5 Mc. This can simplify Eq. (3) to a simple lag, since the break point for the parasitic capacitance pole of the 1-megohm resistor may be as low as 40 kc. In the summing-amplifier transfer function, this dominant input-resistor zero is easily canceled by the similar feedback-resistor pole [Eq. (12)]. Matched capacitors can be added in parallel with summing-amplifier input and feedback resistors for improved high-frequency performance and stability margin (Art. 2.4.14).

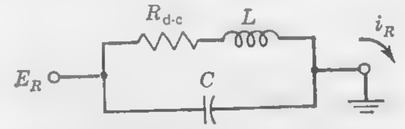


FIG. 2.4.1. Equivalent circuit for a computing resistor.

In *integrators* the parasitic capacitance is harmful and must be kept as low as possible. The 1-megohm integrator *input* resistor with the maximum 4-$\mu\mu$f parasitic capacitance would cause +0.25 per cent maximum amplitude error at 100 cps due to phase shift (Art. 2.4.3) from this one source. With the standard capacitor values of 0.01 to 1 μf, 100 cps computing frequencies can be achieved only with 100-kilohm integrator input resistors. With a similar maximum parasitic capacitance of 4 $\mu\mu$f, the 100-kilohm integrator resistor would cause only +0.025 per cent maximum amplitude error due to phase shift at 100 cps.

Present wire-wound resistor a-c characteristics are acceptable because resistor phase errors essentially cancel for summing amplifiers and are relatively small with the 100-kilohm integrator input resistors normally necessary for high computational frequencies.

INTEGRATING CAPACITORS

2.4.6. Properties of Practical Capacitors. Stability. The practical adjustable capacitor approximates its desired ideal transfer impedance $Z_i(s) = 1/C_i s$ much less closely than a resistor approximates ideal resistance. Capacitor characteristics are less stable, and their idealized equivalent circuits are more complex. Even the nominal *value* of a given capacitor is subject to different interpretations (Art. 2.4.10). The principal reasons for using a resistance-time measurement technique of some sort for capacitor-value determination are: first, convenience, since the same resistance standard used for resistor measurement can be used to form the time constant to be measured, and second, it is difficult to maintain precise capacitance standards for conventional bridge measurement.[6] Also, bridge frequencies usually exceed analog-computing frequencies and hence yield changed capacitance values.

Even carefully selected capacitors in a constant environment show *variation in value with time.* This drift is influenced by the magnitude of recent value adjustments, by the time following the adjustment, and by other undetermined factors. To achieve the best possible stability, it is necessary to maintain the capacitors in a well-controlled environment and to be able to readjust them *without thermal disturbance.*

The temperature coefficient of the computing *capacitors* rather than that of the resistors will determine the component-oven temperature-stability requirement. Care should be taken to see that the capacitors are never severely temperature-cycled; a 10°C drop in temperature may necessitate capacitor readjustment, even though the oven temperature is returned to its exact original value. A maximum *temperature-retrace error* should be specified.

Capacitors change value with voltage level because of mechanical stresses introduced by variation in electrical forces with voltage. This effect can be minimized in manufacturing but should also be specified.

2.4.7. D-C Leakage. D-C leakage limits integrator accuracy at low frequencies. This leakage increases with temperature; this factor imposes an upper limit on component-oven set-point temperature. The leakage resistance is readily detected by the decay rate of large integrator voltages in HOLD, with

$$R_{dc} = \frac{E_{max}}{C} \frac{1}{dE/dt_{max}} \tag{2.4.5}$$

which should be averaged over positive and negative values of E_{max} to cancel the amplifier grid current and offset errors. This measurement should be made only after a few minutes in RESET to avoid dielectric-absorption errors (Art. 2.4.11).

An alternate method involves measurement of amplitude decay in a three-amplifier sine generator below 0.001 cps, where

$$R_{dc} \approx \frac{1}{\alpha C} \tag{2.4.6}$$

Equal capacitors (C) are used in the two integrators, and α is the attenuation constant of exponential decay (Art. 2.4.15).

2.4.8. Dielectric Losses. Choice of Dielectric Material. In practical computer-capacitor applications, various "relaxation mechanisms" (dielectric absorption) dissipate more energy than direct resistive leakage. The dissipation factor is defined as

$$D = \frac{\text{average power loss}}{\text{peak power stored}} \approx \frac{1}{\omega R_1(f) C(f)} \tag{2.4.7}$$

for steady-state sinusoidal capacitor voltage, where $R_1(f)$ is the parallel equivalent-loss resistor and $C(f)$ is the capacitor value (both frequency-dependent). For frequency and material variation see Fig. 2.4.2.

Integrating-capacitor dissipation is an important error source at typical computing frequencies. A three-amplifier sine-generating loop (Fig. 2.4.7) with perfect ampli-

fiers, resistors, and potentiometers would yield a per-radian amplitude decrement equal to the dissipation factor D.

To achieve less than 0.1 per cent amplitude loss per radian of steady-state sine-wave signal, only Mylar, polystyrene, or Teflon dielectric capacitors can be considered. For higher accuracies, only polystyrene or Teflon should be used (Fig.

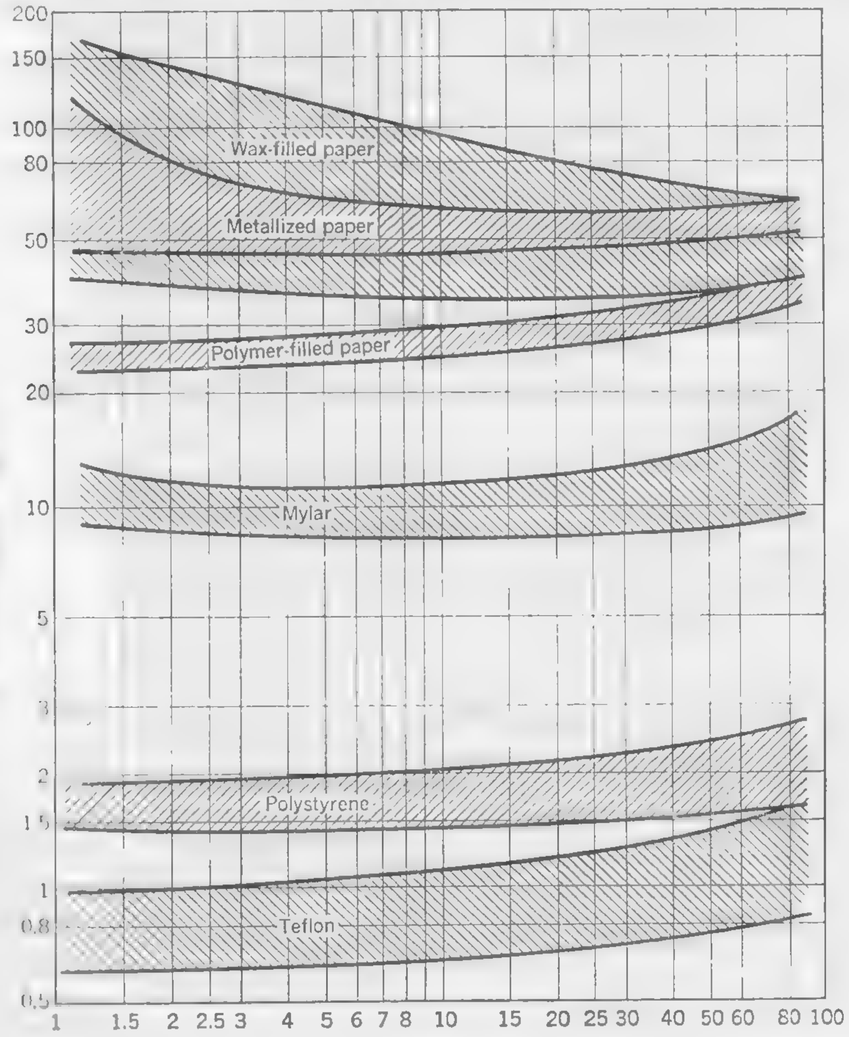


FIG. 2.4.2. Dissipation factor D of 0.1-μf capacitors vs. frequency for various dielectric materials at 25°C.

2.4.3). Teflon cannot as yet be recommended over polystyrene because of the poorer stability characteristics of Teflon (Art. 2.4.6).

2.4.9. Capacitor Equivalent Circuits. The frequency dependence for $C(f)$ in Eq. (7) can generally be neglected (approximately -0.02 per cent per decade frequency), while the parallel equivalent loss resistor varies almost inversely with frequency. Using

$$R_1 \approx \frac{1}{1.5(10)^{-4}\omega C} \tag{2.4.8}$$

for polystyrene dielectric would give a constant D of $1.5(10)^{-4}$, which shows reason-

ably good agreement with the experimental data (Fig. 2.4.3) over the five-decade frequency range. This simple parallel resistor-capacitor circuit is therefore preferred for *steady-state* sinusoidal error analyses, where frequency-dependent parameters are not troublesome.

The linear equivalent circuit of Fig. 2.4.4 approximates the typical polystyrene capacitor with respect to dissipation factor, capacitance variation with frequency,

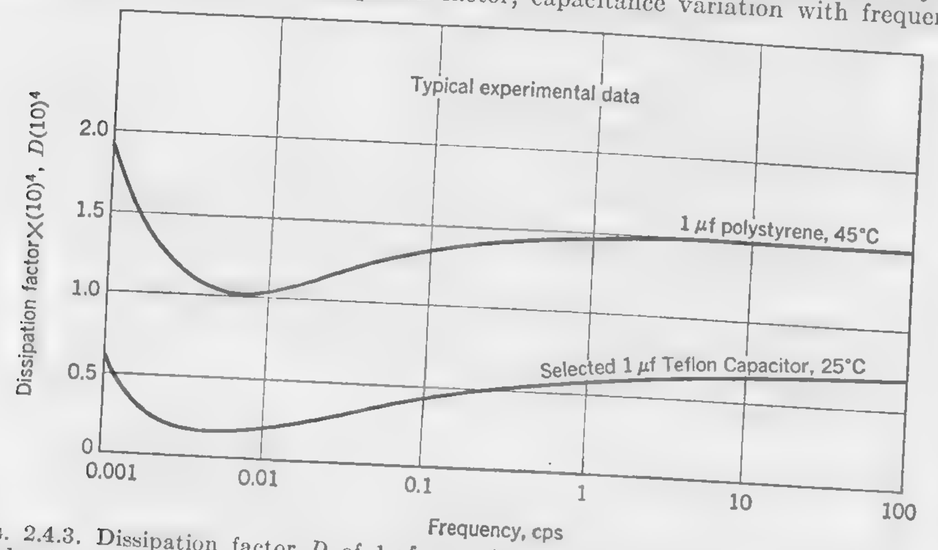


FIG. 2.4.3. Dissipation factor D of 1-μf capacitors vs. frequency. Shown for a typical production polystyrene unit and selected Teflon unit to compare present characteristics with possible future improvements in capacitor electrical purity.

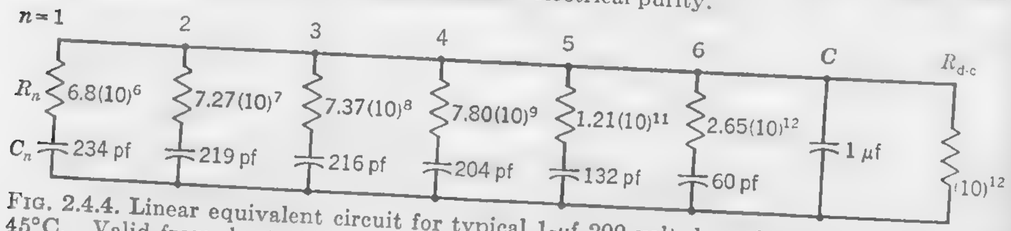


FIG. 2.4.4. Linear equivalent circuit for typical 1-μf 200-volt d-c polystyrene capacitor at 45°C. Valid from d-c to 100 cps for leakage, dissipation factor, capacity variation, and dielectric absorption.

and transient response (Art. 2.4.11). The equivalent transfer impedance is

$$Z_t(s) = \frac{1}{\frac{1}{R_{dc}} + Cs + \sum_{n=1}^{6} \frac{C_n s}{1 + R_n C_n s}} \tag{2.4.9}$$

and the equivalent dissipation factor is

$$D \approx D_{dc} + D_n = \frac{1}{\omega R_{dc} C} + \sum_{n=1}^{6} \frac{C_n}{C} \left[\frac{\omega/\omega_n}{1 + (\omega/\omega_n)^2} \right] \tag{2.4.10}$$

where $\omega_n = 1/R_n C_n$. This equivalent dissipation factor formed the basis of the circuit of Fig. 2.4.4 and shows excellent agreement (±5.5 per cent) with the data of Fig. 2.4.3 and also from zero frequency to 0.001 cps.

A simplified circuit, with a single series resistance and capacitance representing one dielectric-absorption term, has been investigated analytically (see also Art. 2.4.11).[3,5]

2.4.10. Frequency Dependence of Capacitance. Comparing the linear equivalent circuit (Art. 2.4.9) with that of the frequency-dependent simple equivalent circuit

shows that the effective capacity decreases only slightly with frequency, and the equivalent loss resistance decreases approximately linearly with frequency. The effective capacitance variation is approximately −0.02 per cent variation per frequency decade, as indicated by the C_n values in Fig. 2.4.4, whose $R_n C_n$ time constants were selected at decade intervals. Careful measurement of three-amplifier sine-generator oscillation periods confirms this capacitance variation. Equivalent-loss-resistance variation is easily confirmed by dissipation-factor measurement.

A recommended and simple adjustment procedure for setting nominal 1-μf capacitors is to adjust three such capacitors iteratively for correct oscillation period (using two capacitors at a time) at 1 radian per sec.[5] This has the effect of "balancing" the

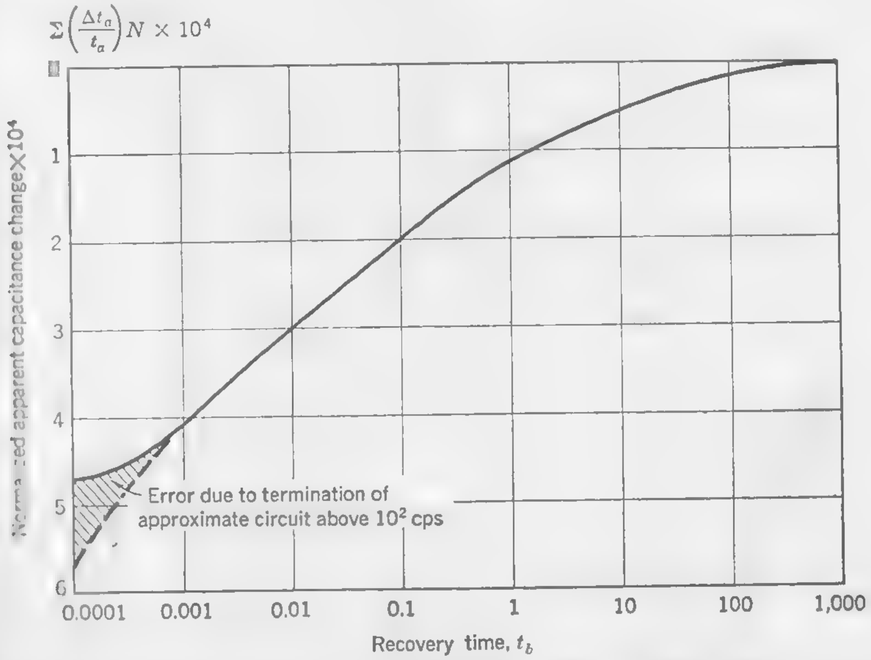


FIG. 2.4.5. Capacitance change vs. recovery time. Ramp-parabola apparent capacitor value changed by initial condition or *recovery time* only. Error due to termination of approximate circuit (Fig. 2.4.4) above $(10)^2$ cps.

capacitor variation over the normal 1-μf analog-computer frequency range (about 0.01 to 1.6 cps). The C value of Fig. 2.4.4 is thereby adjusted to 0.9931 μf; 0.1-μf capacitors are similarly adjusted at $\omega = 10$ radians per sec, which is the correct center frequency for the smaller capacitors.

2.4.11. Transient Effects of Dielectric Absorption. The equivalent circuit of Fig. 2.4.4 was shown to be adequate for dielectric-absorption or capacitor-history phenomena by careful reevaluation of the apparent capacitor-value change with initial condition or recovery-time interval using the transient (ramp-parabola) capacitor measurement method.[5,7] This method yields very consistent apparent capacitor values (±0.0008 per cent) in repeated measurements *if* the recovery time or reset interval between runs is constant. A change in the reset interval will cause the apparent capacitance value to shift to a quite different but equally stable value.

A rigorous analysis of this apparent capacitance-value shift has been completed for the polystyrene equivalent circuit of Fig. 2.4.4.[7] Even for a rather mild transient (capacitor shorted for recovery-time intervals $0 < t_b < 1{,}000$ sec) there is a significant shift in the time t_a required for the integrator parabolic output voltage to equal its ramp input voltage in magnitude, $t_a \approx 2R_i C_f$. The analytical curve (Fig. 2.4.5) essentially agrees with careful experimental data and confirms the transient validity of the equivalent circuit.

In practice, a computer may be placed in HOLD for a considerable time. If the HOLD capacitor voltage differs considerably from its subsequent initial-condition voltage, the subsequent problem solution may be affected considerably more than expected from Fig. 2.4.5; that is, it may approach an apparent error of 0.1 per cent.

COEFFICIENT-SETTING POTENTIOMETERS

2.4.12. Equivalent Circuit for Copper-mandrel Potentiometers. If summing-amplifier network elements are capacitively compensated to improve the system-

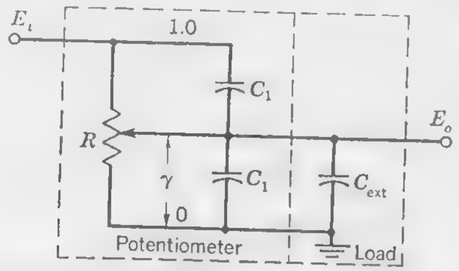


FIG. 2.4.6. Potentiometer equivalent circuit

$$\frac{E_o}{E_i}(s) = \frac{\gamma[(1-\gamma)R_1C_1s + 1]}{(1-\gamma)R[2 + (C_{ext}/C_1)]s + 1}$$

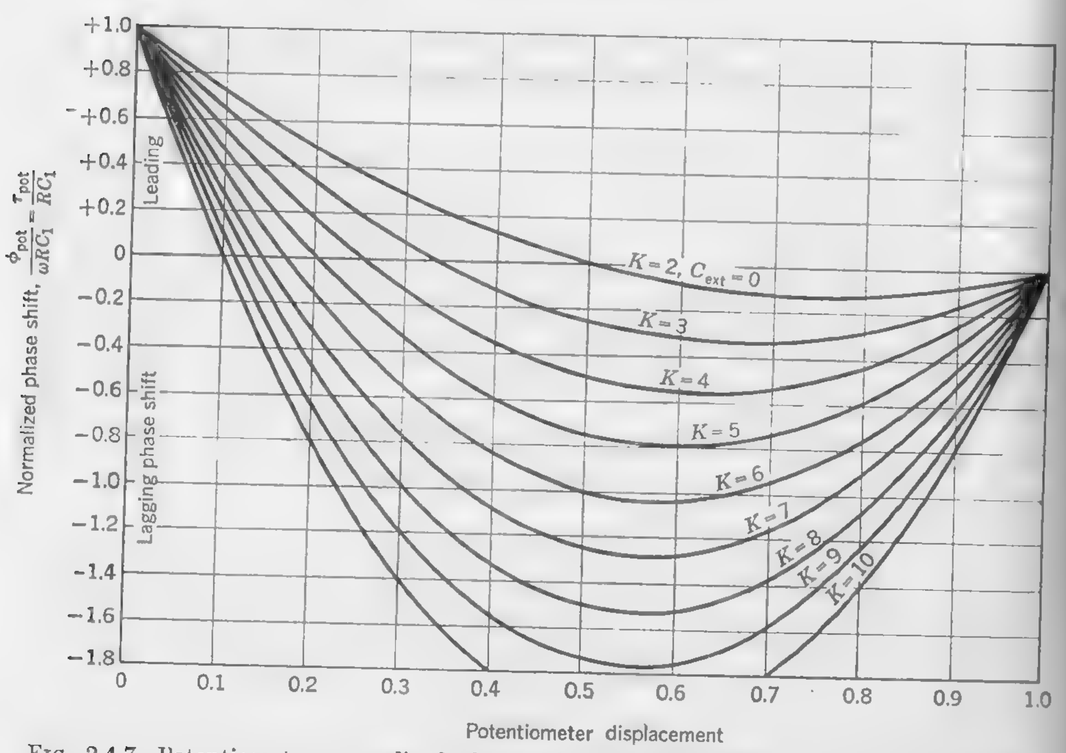


FIG. 2.4.7. Potentiometer normalized phase shift vs. displacement. See Table 2.4.1 for actual f_{max} and C_1 values. Curves valid only for resistive loading, where $R_1 \gg \gamma(1-\gamma)R$, and

$$\frac{\phi_{pot}}{\omega RC_1} = \frac{\tau_{pot}}{RC_1} = (1-\gamma)(1-K\gamma)$$

$$K = \frac{C_1 + C_2}{C_1} = 2 + \frac{C_{ext}}{C_1}$$

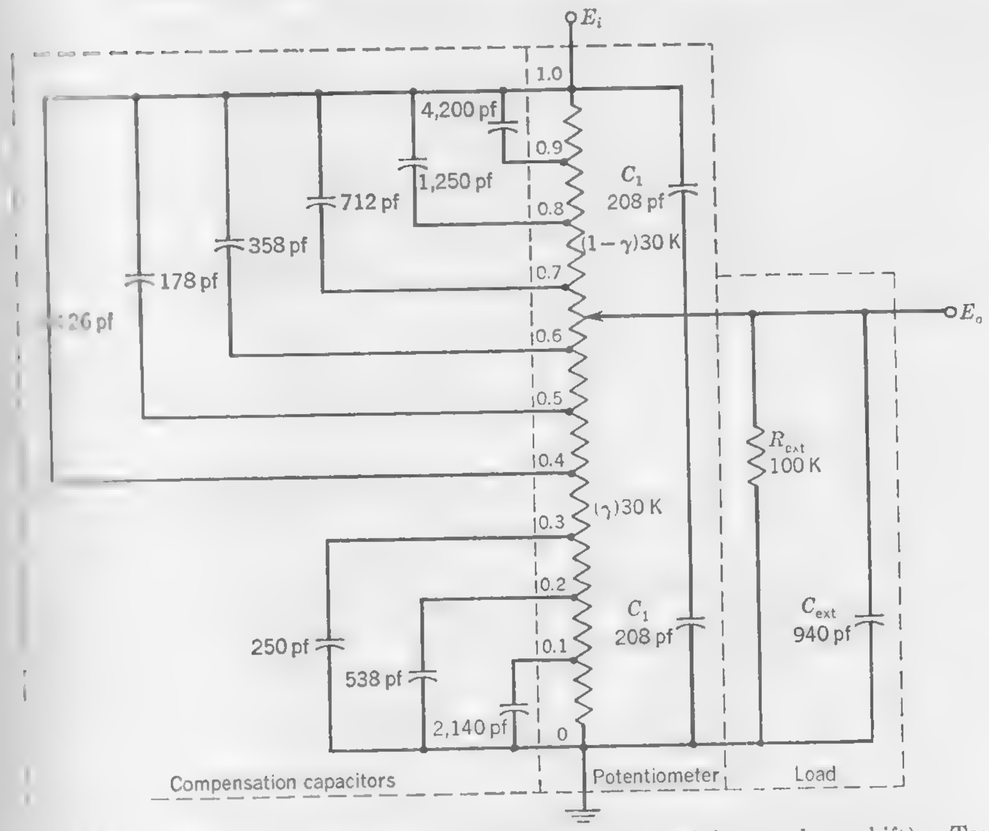


FIG. 2.4.8. Capacitively compensated potentiometer (for minimum phase shift). Tap-capacitor values valid for only 100-kilohm and 940-$\mu\mu$f loading.

stability margin (Art. 2.4.14), the phase shift from potentiometer input to wiper arm can easily become the largest single error source at computing frequencies. Figure 2.4.8 shows an equivalent circuit for the typical computer multiturn copper-mandrel coefficient-setting potentiometer.[8] It is adequate for most analog-computer error analyses. C_1 is essentially constant for all potentiometer settings and experimentally shown to be only slightly dependent on potentiometer value. This equivalent circuit is valid within the limits listed in Table 2.4.1.*

Table 2.4.1

R, kilohms	C_1, $\mu\mu$f	Equivalent-circuit frequency limit, cps	Fig. 2.4.7 frequency limit, cps
10	200	$1.1(10)^6$	$6.6(10)^3$
20	204	$550(10)^3$	$3.3(10)^3$
30	208	$350(10)^3$	$2.1(10)^3$
50	216	$200(10)^3$	$1.2(10)^3$
100	224	$82(10)^3$	590

The equivalent-circuit frequency limit is based on experimental step-function data, while the frequency limit of Fig. 2.4.7 is based on a first-order low-frequency mathe-

* Data for type A 10 turn potentiometers manufactured by Beckman/Helipot Div.

matical equivalent of Fig. 2.4.6, where the potentiometer phase shift is given by

$$\phi \approx \omega\tau_{\text{net}} = RC_1(1 - \gamma)(1 - K\gamma) \qquad (2.4.11)$$

and

$$K = 2 + C_1/C_{\text{ext}}$$

The low-frequency phase shift is invariably the most important error source at computing frequencies (Art. 2.4.3). It has therefore been plotted in normalized form, $\phi/\omega RC_1$, in Fig. 2.4.7 for various values of K to allow ready determination of potentiometer phase shift for any potentiometer value, frequency, displacement, and load capacitance.

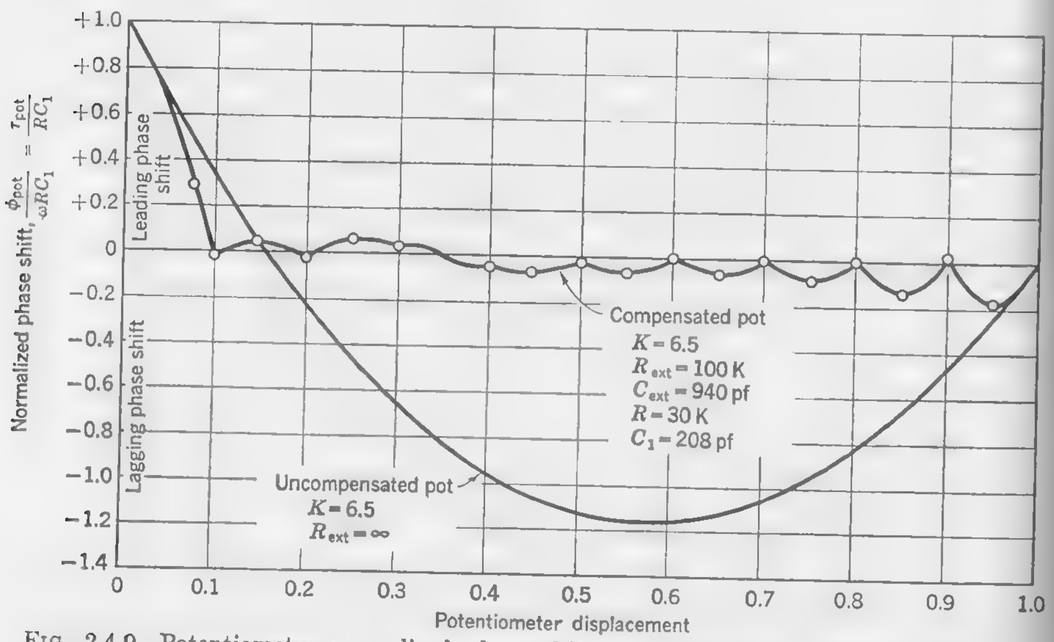


FIG. 2.4.9. Potentiometer normalized phase shift vs. displacement, showing effect of potentiometer compensation.

$$\frac{\phi_{\text{pot}}}{\omega RC_1} = \frac{\tau_{\text{pot}}}{RC_1} \qquad \text{(for both potentiometers)}$$

$$= (1 - \gamma)(1 - K\gamma) \qquad \text{(for uncompensated potentiometer only)}$$

$$K = \frac{C_1 + C_2}{C_1} = 2 + \frac{C_{\text{ext}}}{C_1}$$

2.4.13. Capacitive Compensation of Copper-mandrel Potentiometers. From the simple equivalent circuit (Art. 2.4.12), it is easy to compute a compensating capacitor to be connected between potentiometer wiper and either potentiometer input or ground so as to cancel phase error for each specific combination of potentiometer setting and resistive-capacitive loading.

A more practical scheme employs capacitors connected to potentiometer taps for approximate compensation at any potentiometer setting (Fig. 2.4.8).[9] The experimental normalized phase shift is greatly improved (Fig. 2.4.9), and the square-wave response is still excellent (Fig. 2.4.10) at 25,000 cps; note that the low-frequency phase error is the more important criterion for analog-computer use. A few such potentiometers will occasionally prove invaluable for critical simulations in even moderate-sized computer installations.

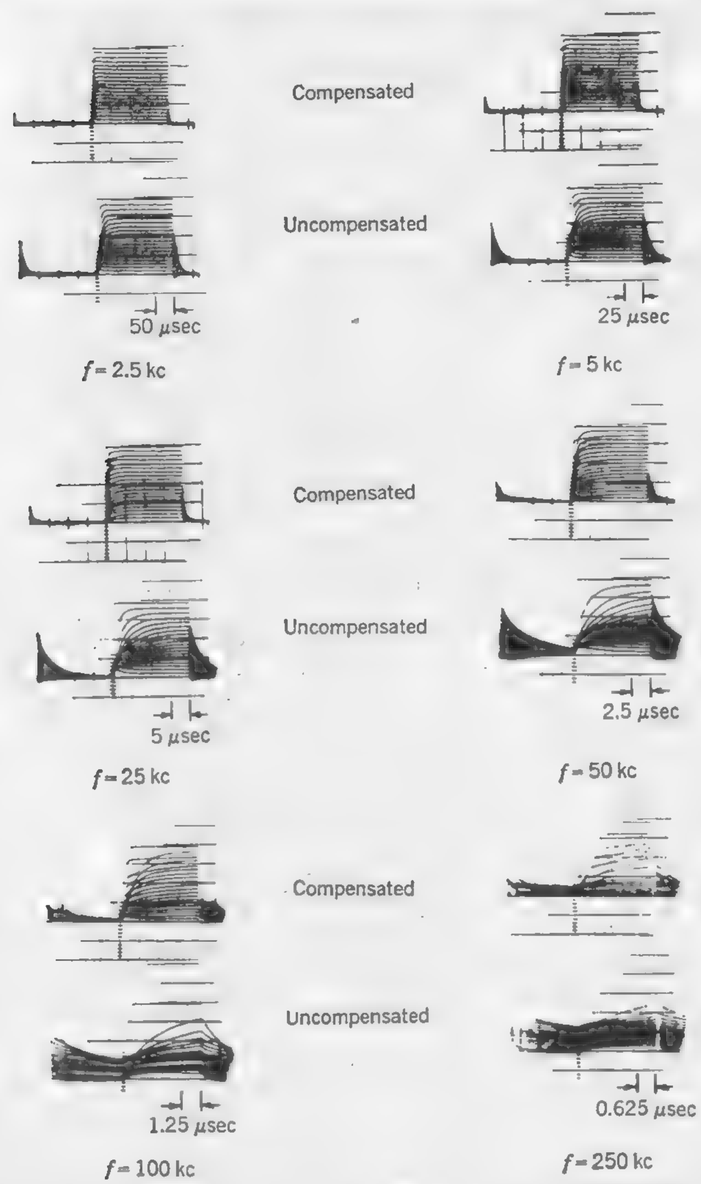


FIG. 2.4.10. Oscillograms of square-wave response for copper-mandrel potentiometers, showing effect of capacitive compensation. $R = 30$ kilohms. $C_{\text{ext}} = 940$ $\mu\mu$f. Displacement γ in 5 per cent increments. Compensated 50-kc response ≈ uncompensated 25-kc response.

COMBINED OPTIMIZATION OF LINEAR COMPUTING ELEMENTS

2.4.14. Capacitive Compensation of Summing Amplifiers. Referring to Sec. 2.3, it is desirable to hold the operational-amplifier-system equation

$$\frac{E_o}{E_i}(s) = \frac{Z_f}{Z_i}\left[\frac{\dfrac{G(s)Y_fY_0}{Y_aY_b - Y_f^2}}{1 + \dfrac{G(s)Y_fY_0}{Y_aY_b - Y_f^2}}\right]\left[1 - \frac{Y_f}{G(s)Y_0}\right] \qquad (2.4.12)$$

closely identical to the ideal summing-amplifier equation

$$\frac{E_o}{E_i}(s) = -\frac{Z_f}{Z_i} \approx -\frac{R_f}{R_i} \tag{2.4.13}$$

over as wide a frequency range as possible. The necessity of balancing Z_f/Z_i in an a-c sense in addition to the normal d-c sense has been discussed in Art. 2.4.5. With capacitively matched resistors, the system root-locus plot of a typical operational amplifier is shown in Fig. 2.4.11*a*. The very-low-frequency closed-loop poles and zeros at the right-hand side of the diagram have very little bearing on either the high-frequency stability characteristic or low-frequency computational accuracy. However, the pole caused essentially by the summing-junction capacity (actually Y_a) combines with the deliberate low-frequency pole to yield a dominant second-order system characteristic. This is shown by the location of the closed-loop poles in the lower part of Fig. 2.4.11*a*. This system characteristic is typical of most operational amplifiers in use and results in the usual peaking and rapid phase-shift increase of a second-order system. The bandwidth for such an arrangement is between 15 and 20 kc, and the stability margin is sufficient only for standard circuity configurations (not for low feedback impedance, limiters, driving remote equipment, etc.).

It has long been known that a fairly large capacitor placed around the feedback resistor would stabilize the amplifier at the expense of bandwidth and computing-frequency phase error (a standard technique now used to achieve stable limiters, etc.). If, however, an *exactly* corresponding capacitor is placed around the input resistor, the original favorable low-frequency phase-shift characteristics can be retained *while the bandwidth is increased and the system-stability margin is considerably improved.* The root-locus diagram of Fig. 2.4.11*b* shows the effect of capacitive compensation of both feedback and input resistors. There is a slight movement (to the right) of the pole caused by Y_a but a considerable movement (again to the right) of the zero caused by Y_f. This means that the zero due to Y_f may be close enough to bring the root locus back to the real axis. This is the case for typical amplifier gains, and the resultant closed-loop poles are not only at a much higher frequency but also have the more desirable first-order closed-loop roll-off characteristic.

As an example, the Berkeley EASE 1048B operational amplifier had a bandwidth of approximately 13 kc with a peaking of about 3 db. This same amplifier with capacitively compensated resistors produces the bandwidth characteristic shown in Fig. 2.4.12. Note that the roll-off characteristic of this circuit differs considerably from the more typical underdamped quadratic characteristic and yields a much higher system-stability margin.

With three such amplifiers in a loop, the amplitude margin is three times the amplitude margin of one amplifier at 60-deg phase shift; 60-deg phase shift occurs at approximately 90 kc for this amplifier. Under these conditions there is approximately 3.8 db gain margin from each amplifier, or 11.5 db gain margin for the loop. Hence the loop gain can be increased to approximately 5.7 before oscillation starts; this compares with a loop gain of about 2 for typical uncompensated network resistors and amplifiers. This is a significant improvement for either differential- or algebraic-equation loops of high gain.

This technique is of general interest since *it can be applied to existing computers to improve their transient response, bandwidth, and system-stability margin.* Another indication of the improved stability margin is the ability of summing amplifiers to drive heavy capacitive loads. With the recommended 40-$\mu\mu$f (1-megohm) compensation, the damping factor ζ (assuming a quadratic response) still exceeds 0.2 with 0.1-μf capacitive load. A compensated-resistor time constant larger than 40 μsec generally reduces the stability margin.

Another undesirable effect resulting from too much capacitive compensation on the input resistors of summing amplifiers is due to capacitive loading of a preceding potentiometer. In particular, if a summing amplifier has both 1-megohm and 100-kilohm summing resistors (gains of 1 and 10), capacitive compensation of the 1-megohm resistors with 100 $\mu\mu$f would require 0.001 μf for the 100-kilohm resistors,

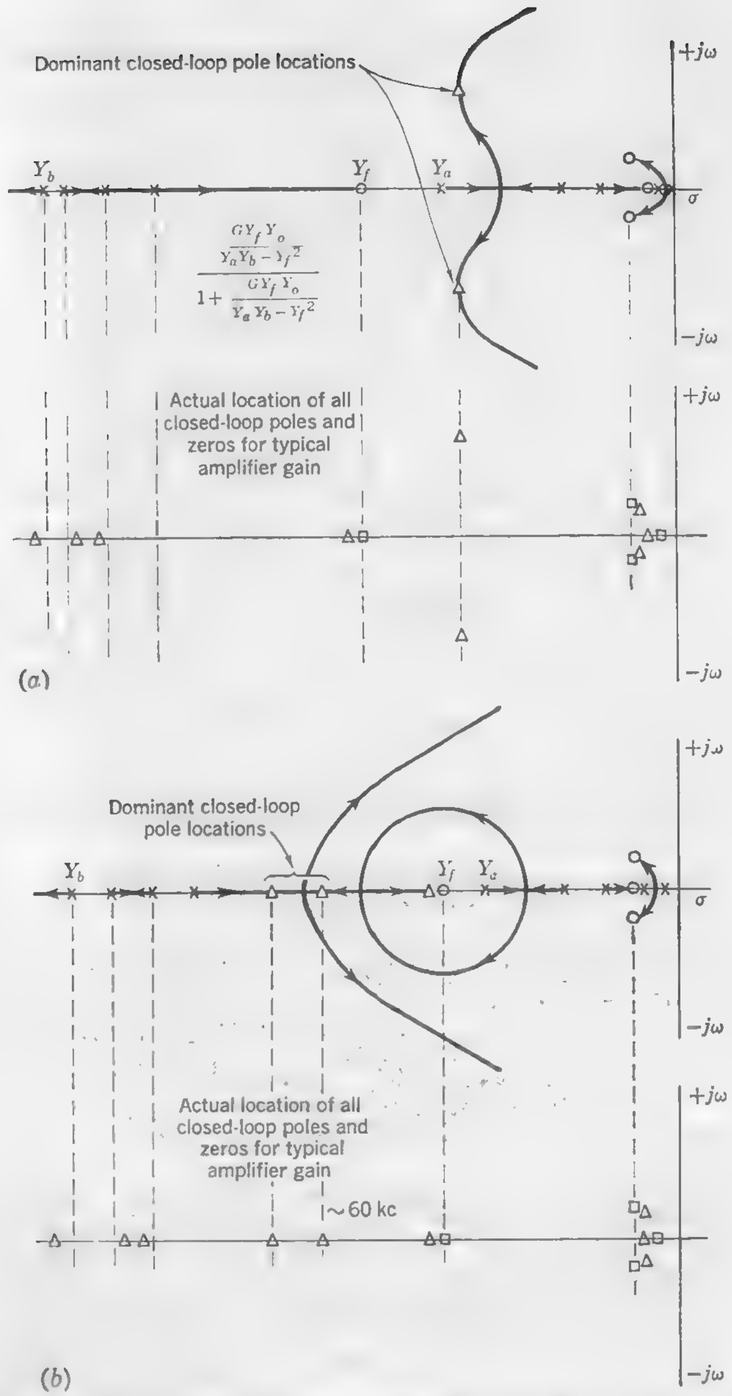


Fig. 2.4.11*a*. System root locus of typical summing amplifier (not to scale). Note the dominant second-order characteristic. ×, open-loop pole; Δ, closed-loop pole; ○, open-loop zero; □, closed-loop zero.

Fig. 2.4.11*b*. System root locus of capacitively compensated summing amplifier (not to scale). Note that the dominant poles are *real*.

which would combine with the cabling capacitance to cause excessive potentiometer phase shift (Fig. 2.4.7). This situation can be improved by providing a few special potentiometers phase-compensated for these loads (Art. 2.4.13).[9]

If compensated resistors are used in a summing amplifier, *care must be taken to remove the compensating capacitors from the input resistors if the amplifier is ever converted to an integrating amplifier.*

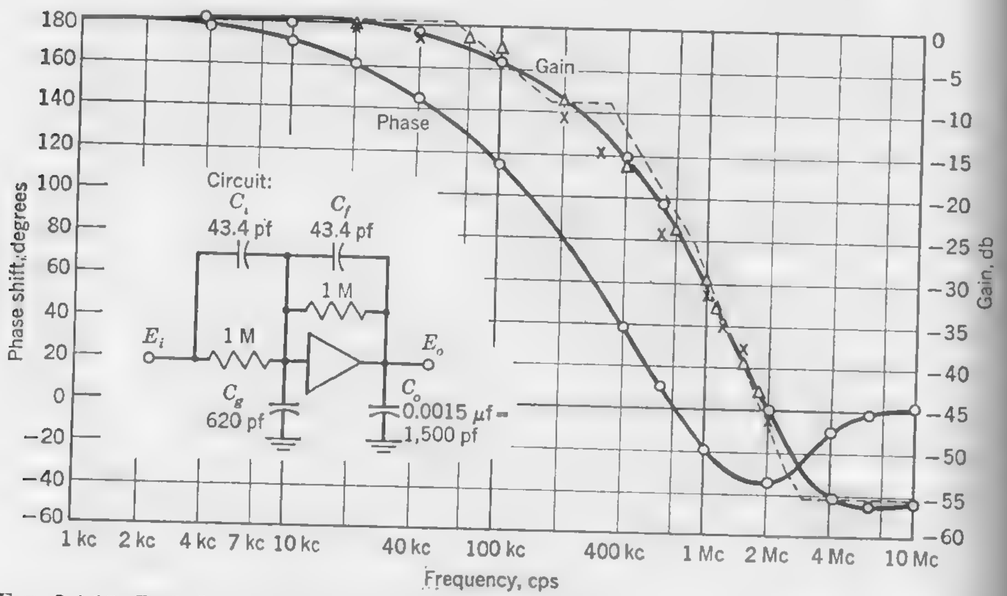


FIG. 2.4.12. Berkeley 1048B frequency response (analytic vs. experimental data). Δ, × experimental data; ⊙, analytical data based on above circuit with $G(s)$ of Berkeley 1048B amplifier approximated by

$$G(s)\ \frac{5.712(10)^{13}[s + 1.1(10)^6]}{s[s + 2(10)^6][s + 5(10)^6]}$$

resulting in

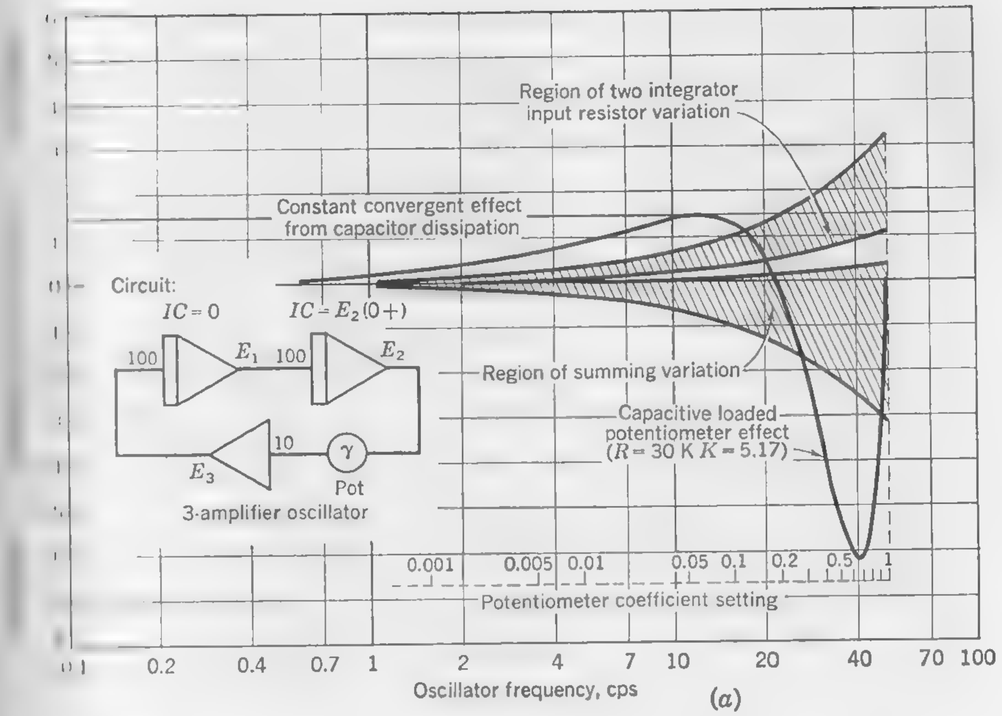
$$\frac{E_o}{E_i}(s) = \frac{1.727(10)^{-3}[s + 2.304(10)^4][s + 1.099(10)^6][s - 1.557(10)^7]}{[s + 2.440(10)^4][s + 3.740(10)^5][s^2 + 3.544(10)^6 s + 4.467(10)^{12}]}$$

$$\omega = 1.839(10)^7 \quad \zeta = 0.5843 \qquad \omega = 2.113(10)^6 \quad \zeta = 0.8385$$

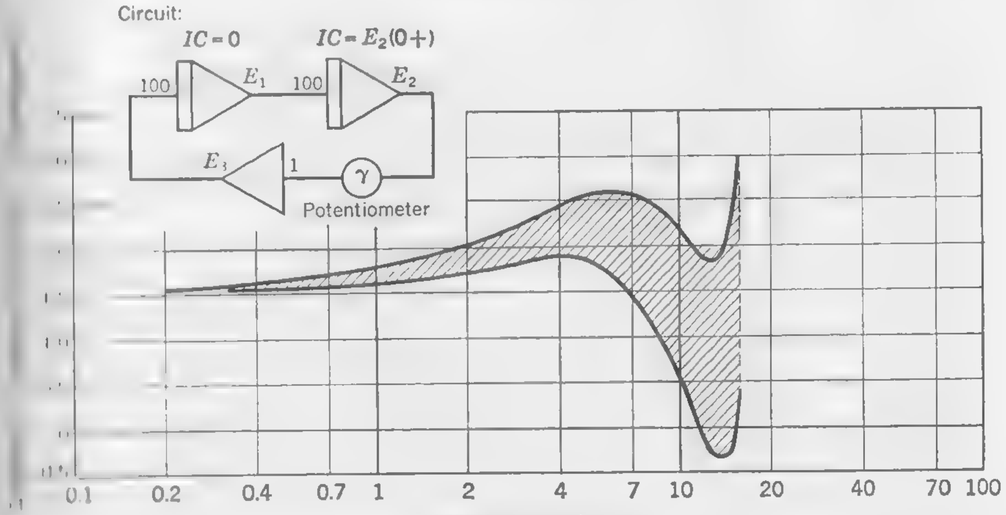
$$\frac{[s^2 + 2.149(10)^7 s + 3.383(10)^{14}]C_i[s + (10)^{-6}/C_i]}{[s + 5.653(10)^6][C_f s + (10)^{-6}/C_f]}$$

2.4.15. System Performance for a Sine-generating Loop.[10] Figure 24.13*a* illustrates a sine-generating loop used to check the performance of the linear computing elements. The components used would be normally 0.1-μf capacitors and 100-kilohm input resistors, giving a gain of 100 volts per sec per volt input for both integrators. With a summing amplifier of gain 10, this will yield over-all loop gains up to $(10)^5$, which results in sine-wave oscillations at a maximum of approximately 50.4 cps. The oscillation would be constant in amplitude with ideal computing elements.

The damping factor due to the capacitor alone equals the capacitor dissipation factor, which for polystyrene is approximately constant at $+1.5(10)^{-4}$. The residual integrator input-resistor lead causes a slight convergent effect as shown, and the summing amplifier can cause either a lead or lag effect depending on resistor a-c match. The remaining important error source is the potentiometer. The data shown (Fig. 2.4.13*a*, *b*) are for a potentiometer capacitive load of approximately 660 μμf, which is approximately 30 per cent less than that encountered in a typical large analog com-



(*a*)



(*b*)

FIG. 2.4.13*a*. Sine-wave damping, $\zeta = \alpha/\omega_n$ vs. frequency (showing individual error sources and expected variation). Equations are

$$E_1(t) \approx E_{1(\max)}\epsilon^{-\alpha t}\sin\omega_n t \qquad E_3(t) \approx E_2(0+)\epsilon^{-\alpha t}\cos\omega_n t$$

$$\omega_n = 316.2\sqrt{\gamma} \qquad \zeta = \frac{\alpha}{\omega_n} \approx -\ln\frac{E_{T1}/E_{T0}}{\omega_n(T_1 - T_0)} \approx D - \frac{(\omega_n \tau_{\text{net}})}{2}$$

where D = capacitor dissipation factor
$\tau_{\text{net}} = \Sigma\tau$'s of each element

FIG. 2.4.13*b*. Sine-wave damping, $\zeta = \alpha/\omega_n$ vs. frequency (showing anticipated system variation). Includes capacitor dissipation, maximum integrator error and variation, maximum amplifier error and variation, and potentiometer error (R = 30 kilohms, K = 3.22). The circuit equations are

$$E_1(t) \approx E_{1(\max)}\epsilon^{-\alpha t}\sin\omega_n t \qquad E_2(t) \approx E_2(0+)\epsilon^{-\alpha t}\cos\omega_n t$$

$$\omega_n = 100\sqrt{\gamma} \qquad \zeta = \frac{\alpha}{\omega_n} \approx -\ln\frac{E_{T1}/E_{T0}}{\omega_n(T_1 - T_0)} \approx D - \frac{(\omega_n \tau_{\text{net}})}{2}$$

puter. It should be noted that the divergent region is caused essentially by the potentiometer phase lag at mid-range potentiometer settings. The "crossover" frequency for this particular case is quite high (approximately 23 to 33 cps). The system is *not* optimized on the basis of this extremely high frequency case, but rather for the more common lower-frequency case shown in Fig. 2.2.13*b*. Here the loop gain is reduced to 10^4, resulting in a maximum frequency of 15.9 cps and a *balanced* error characteristic. Note that the spread in possible damping is mainly from the summing-amplifier resistor mismatch and, as with the potentiometer error, has approximately the *same magnitude* as the capacitor polystyrene dissipation effect, or a maximum of 1.5×10^{-4}, showing a *balance in error effects from all sources.*

2.4.16. Summary. A detailed review of the individual characteristics of the linear elements of the analog computer has been given. Criteria for balanced system optimization have been proposed. As an indication of possibilities: if the inverting amplifier of the high-frequency sine generator of Fig. 2.4.14*a* were selected for a-c matched input and feedback components, the area of possible sine-wave damping would be reduced essentially to a line. With compensated potentiometers, the potentiometer-error effect would disappear (Fig. 2.4.9). This would permit generation of sine waves with a constant damping factor determined primarily by the dissipation factor of the capacitors used. Such system performance has been proved experimentally and can be obtained when needed.

References

1. MacNee, A. B.: Some Limitations on Accuracy of Electronic Differential-equation Solvers, *Proc. IRE*, **40**:303 (1950).
2. Marsocci, V. A.: An Error Analysis of Electronic Analog Computers, *IRE Trans. PGEC*, December, 1956.
3. Dow, P. C.: An Analysis of Certain Errors in Electronic Differential Analyzers, I—Bandwidth Limitations, II—Capacitor Dielectric Absorption, *IRE Trans. PGEC*, December, 1957, and March, 1958.
4. Miura, T., and M. Nagata: Theoretical Considerations of Computing Errors of a Slow Type Electronic Analog Computer, *IRE Trans. PGEC*, December, 1958.
5. Single, C. H.: Precision Components for Analog Computers, *ISA Convention Record*, Paper 56-21-2, September, 1956.
6. Meilander, W. C., and B. H. Hellman: A Technique for Absolute Measurement of Analog Computer Capacitors, *Goodyear Aircraft Tech. Publ.* GER-9075, November, 1958.
7. Brussolo, J., and C. H. Single: Transient Analysis of Capacitor Equivalent Circuit, *Beckman/Berkeley Eng. Rept.* CRD59-5.
8. Single, C. H.: Equivalent Circuit for Copper-mandrel Potentiometers, available from Beckman/Berkeley Div., Richmond, Calif.
9. Single, C. H.: Phase Compensation of Copper-mandrel Potentiometers, *Beckman/Berkeley Eng. Rept.* CRD59-3.
10. Billinghurst, E., and C. H. Single: Extending the Bandwidth of Precision Analog Systems, presented at the National Simulation Council Meeting, Dallas, Tex., October, 1958, available from Beckman/Berkeley Div., Richmond, Calif.
11. Billinghurst, E.: Operational Amplifier Transfer Function, available from Beckman/Berkeley Div., Richmond, Calif.

PART 5: EXAMPLES OF COMPUTER AMPLIFIER AND POWER-SUPPLY CIRCUITS

By GRANINO A. KORN

2.5.1. Examples of Computer Amplifier Circuits (see also Sec. 2.3). Table 2.5.1 lists manufacturer's specifications for a number of American-made d-c amplifiers suitable for medium and large electronic analog computers. Only vacuum-tube amplifiers capable of supplying at least 12 ma at ±100 volts are listed. All but two amplifiers use 60-cps choppers. Circuits are shown in Figs. 2.5.1 to 2.5.6.

Table 2.5.1

Model	D-C gain	Output current (d-c) at ±100 volts	Max d-c offset after balance (referred to input)	Noise (referred to input)	Small-signal frequency response as unity inverter with 1-megohm resistors (no load unless indicated)†	Overload frequency for 100 sin $2\pi ft$ (no load unless indicated)‡	Power supplies required, volts
Beckman/Berkeley Model 1048B	10^8	20 ma	100 μv	10 mv peak-to-peak	down 3 db at 70 kc; 0.07-deg phase shift at 100 cps (in computer)‡	1.5 kc (in computer)	±250 −465
Beckman/Berkeley Model 1148	2×10^8	25 ma	20 μv	8 mv peak-to-peak	down 3 db at 30 kc; 0.07-deg phase shift at 100 cps (in computer)‡	1.5 kc (in computer)	±250 −465
Computer Systems, Inc., Model MC-508-C	10^8	30 ma	50 μv	1 mv rms	down 3 db at 25 kc; 0.2-deg phase shift at 500 cps (4 kilohms resistive load)	1.5 kc (4 kilohms load)	±270 ±100 ±500
Donner Scientific Company, Model 3101	5×10^7	13 ma*	100 μv	2 mv rms	down 3 db at 50 kc; 0.2-deg phase shift at 1,000 cps	450 cps	±200 −416
Electronic Associates, Inc., Model 6.002§	1.4×10^8	30 ma	60 μv	3 mv peak-to-peak (1 mv to 200 cps)	down 3 db at 30 kc; 0.1-deg phase shift at 100 cps		±300 ±110 −500
Goodyear Aircraft Corp., Model GEDA A-14¶	10^8	25 ma	50 μv	10 mv rms (2 mv peak-to-peak to 200 cps)	down 3 db at 20 kc; 0.01-deg phase shift at 100 cps	3 kc	±200 −450
Reeves Instrument Corp., Model A-400-1	50×10^6	21 ma	250 μv	4 mv rms (in computer)	down 3 db at 10 kc; 0.2-deg phase shift at 500 cps (in computer)	1.2 kc (no load); 900 cps (10 kilohms load)(in computer)	±200 ±400

* More powerful model in preparation.
† Depends radically on summing-point and load capacitances and on capacitor compensation of resistors.
‡ With 45 μμf across both 1-megohm resistors.
§ Uses 94-cps chopper.
¶ Commutator stabilized.

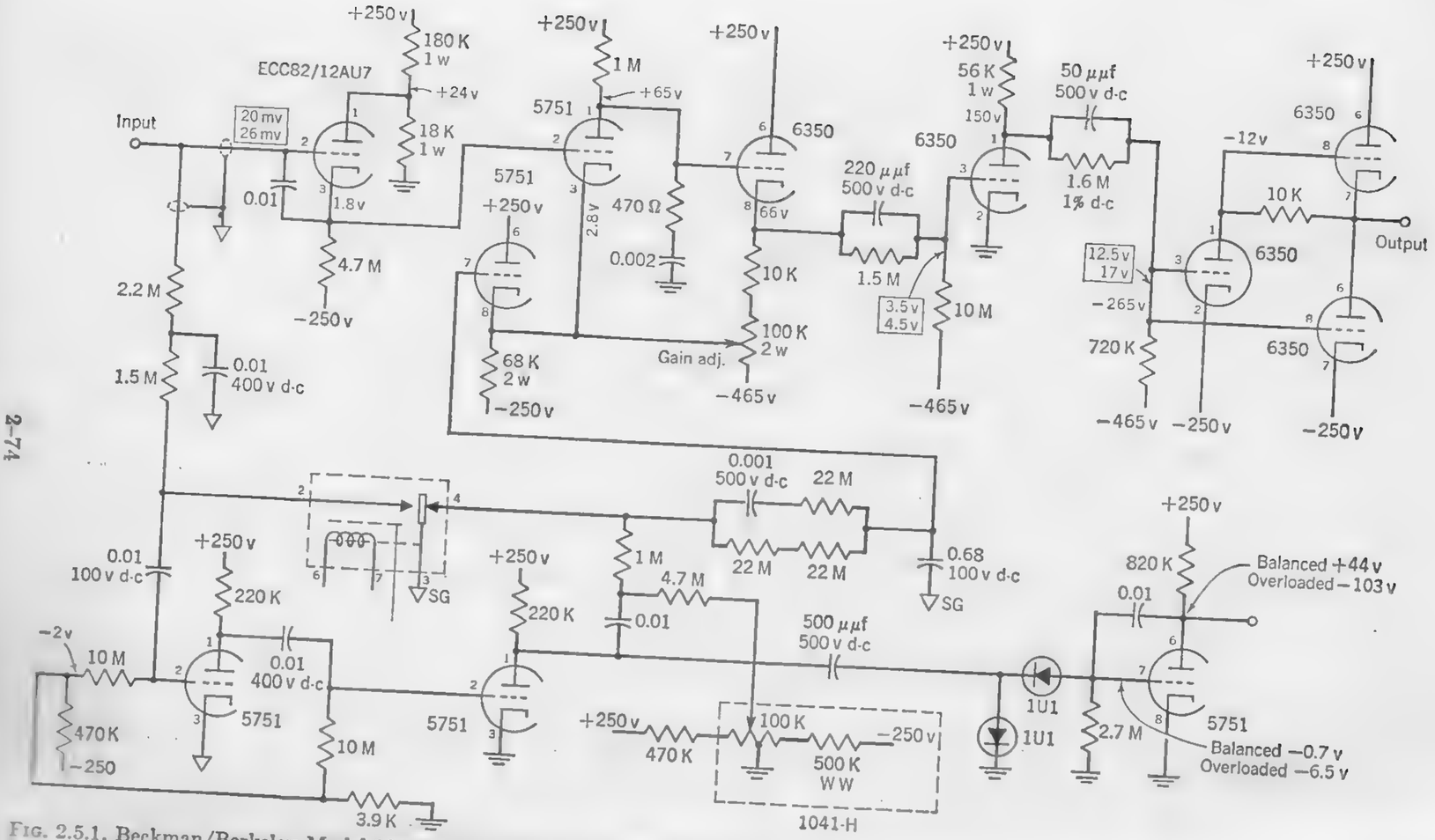


FIG. 2.5.1. Beckman/Berkeley Model 1048B d-c amplifier. A cathode-follower input stage operated at low plate voltage and current yields very low grid current; no grid-blocking capacitor is used. Note the regenerative amplifier stage (see also Sec. 2.3). Boxed voltages are peak-to-peak [illegible] peak-to-peak output without load (upper figure) and with 5-kilohm load (lower figure).

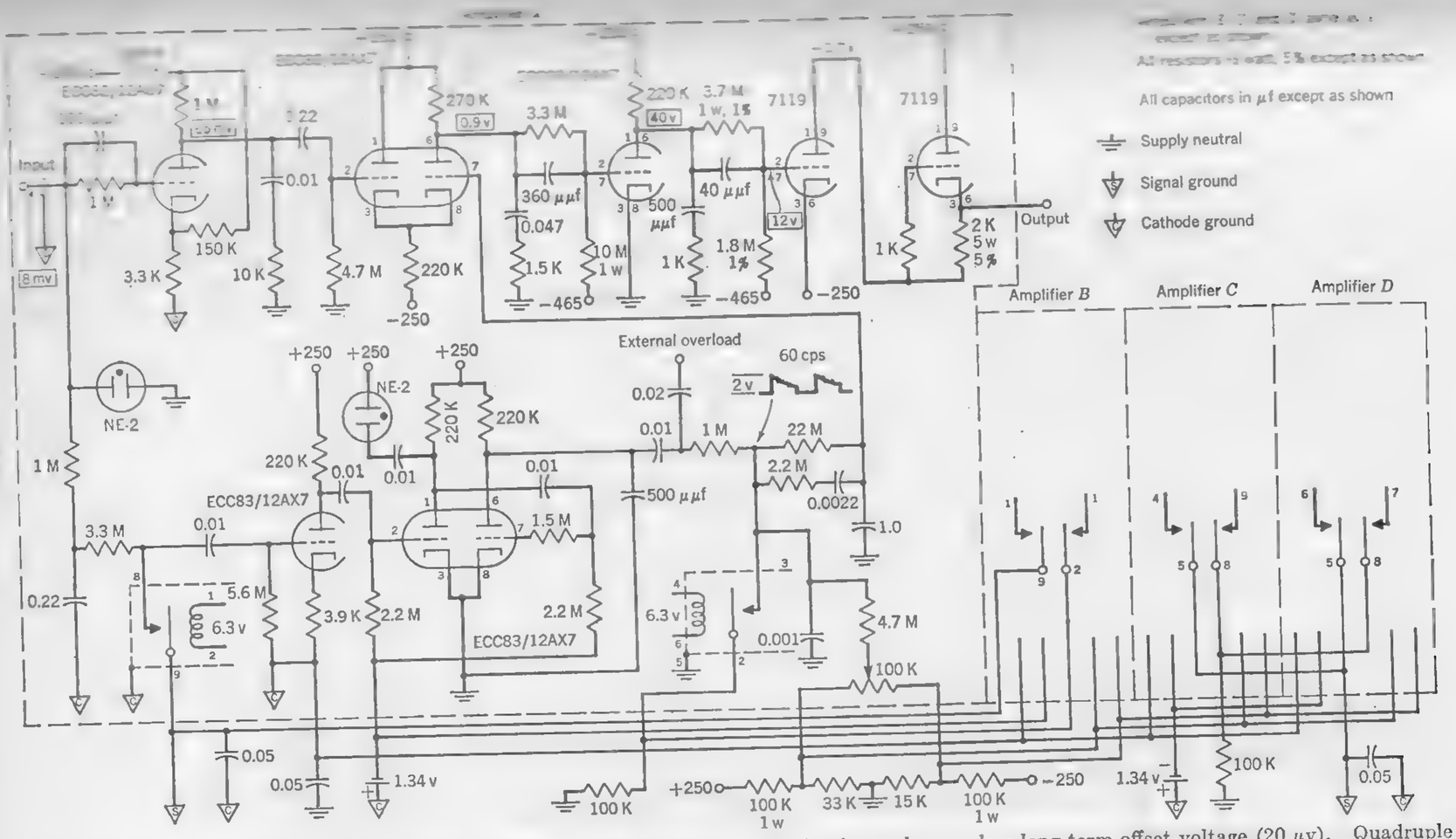


FIG. 2.5.2. Beckman/Berkeley Model 1148 d-c amplifier. High stabilizer-channel gain produces a low long-term offset-voltage (20 μv). Quadruple unit. Boxed voltages are peak-to-peak 100-cps voltages for 200-volt peak-to-peak output without load.

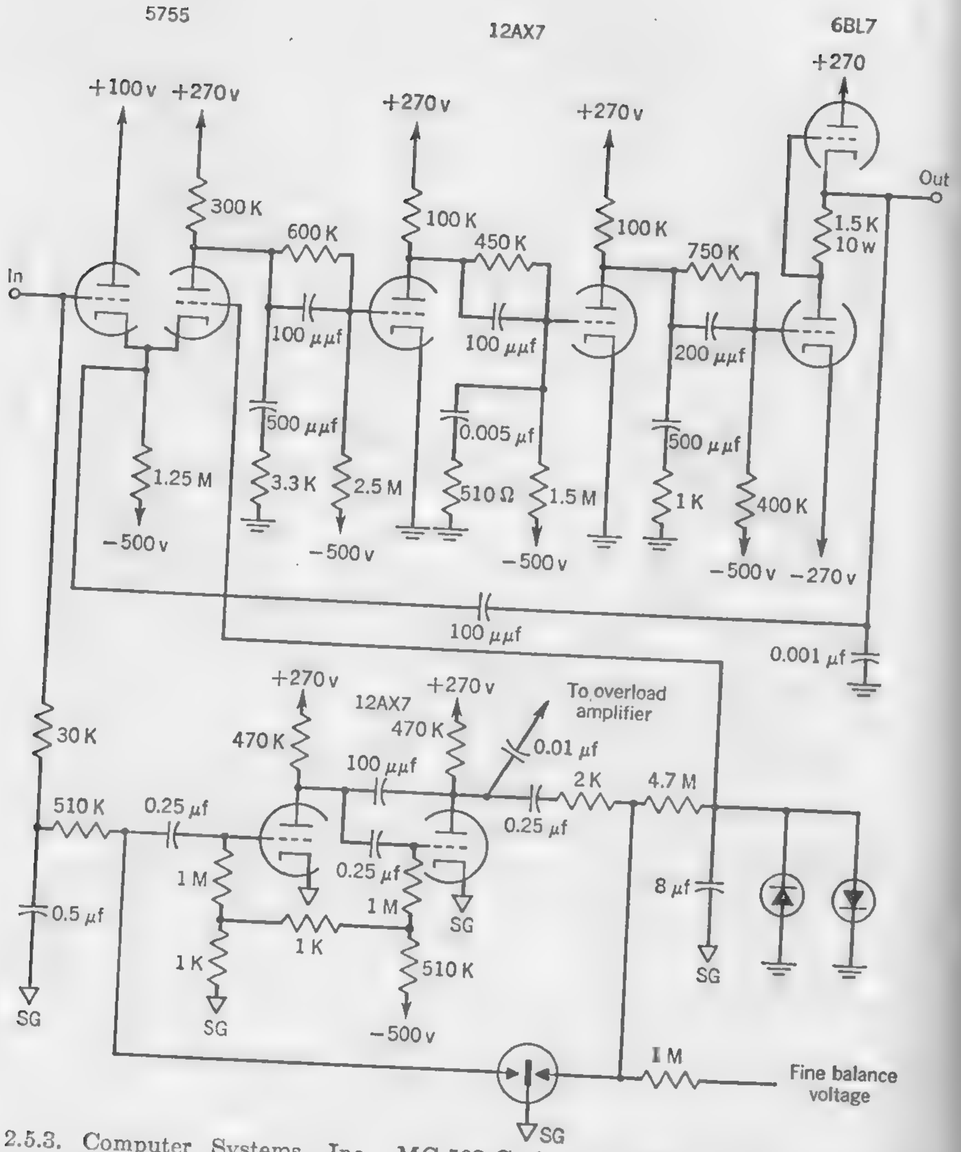


FIG. 2.5.3. Computer Systems, Inc., MC-508-C d-c amplifier. A powerful totem-pole output stage supplies 30 ma at ±120 volts (d-c). Careful equalization yields good high-frequency response (Table 2.5.1). No grid-blocking capacitor is used; the input-stage grid current is less than 10^{-9} amp. Double sets of chopper contacts and removal of chopper excitation in STANDBY increase reliability. 8-hr offset less than 50 μv; noise level less than 1 mv rms with 1-megohm input resistor. Overload recovery time less than 0.5 sec; note overload-discharge diodes. Dual unit; no printed circuits are used.

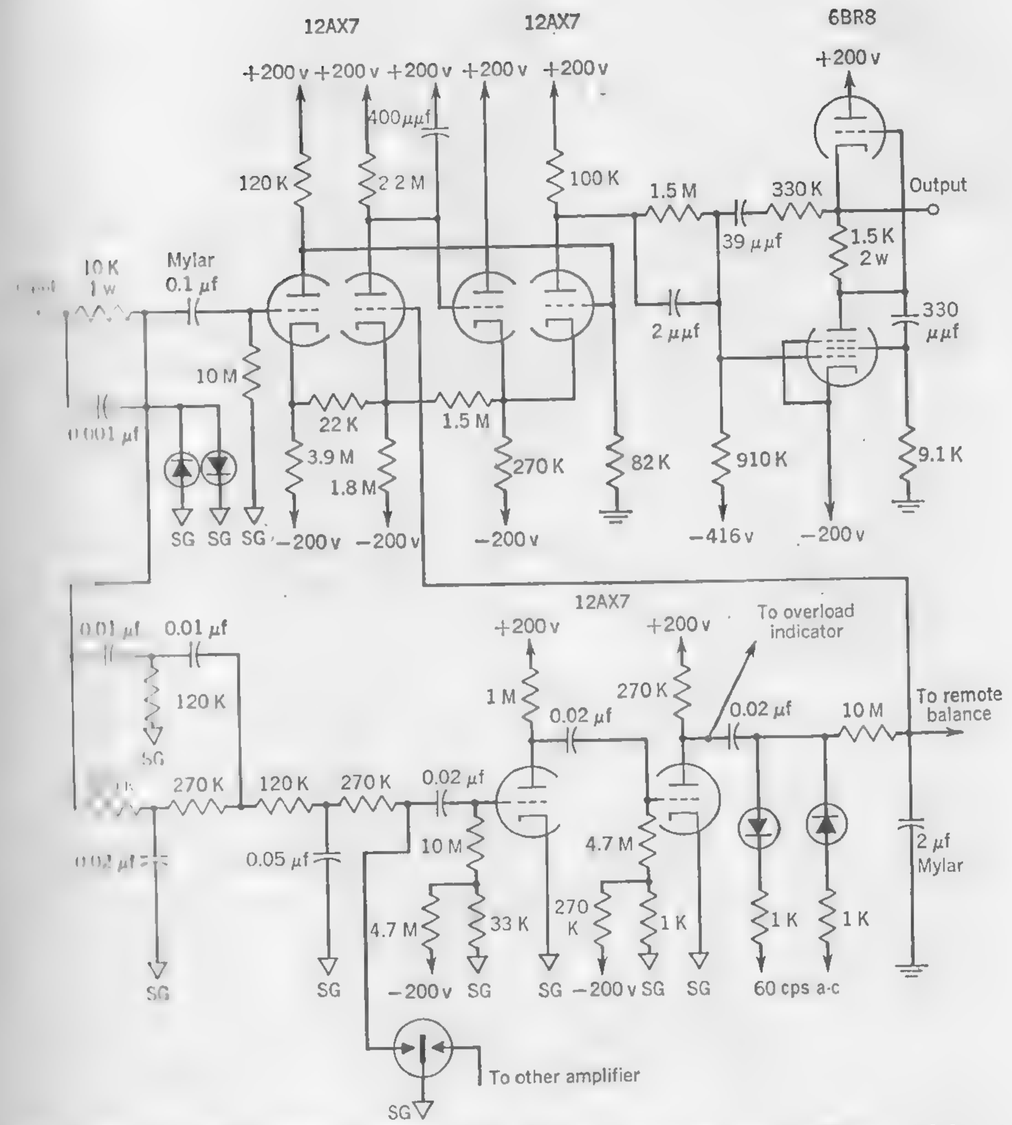


FIG. 2.5.4. Donner Scientific Co. Model 3101 regenerative d-c amplifier. Elaborate filters remove stray 60-cps voltages from the stabilizer input circuit. Semiconductor diodes protect the chopper contacts during overloads. Dual unit; diode demodulators permit one input chopper to serve two amplifiers.

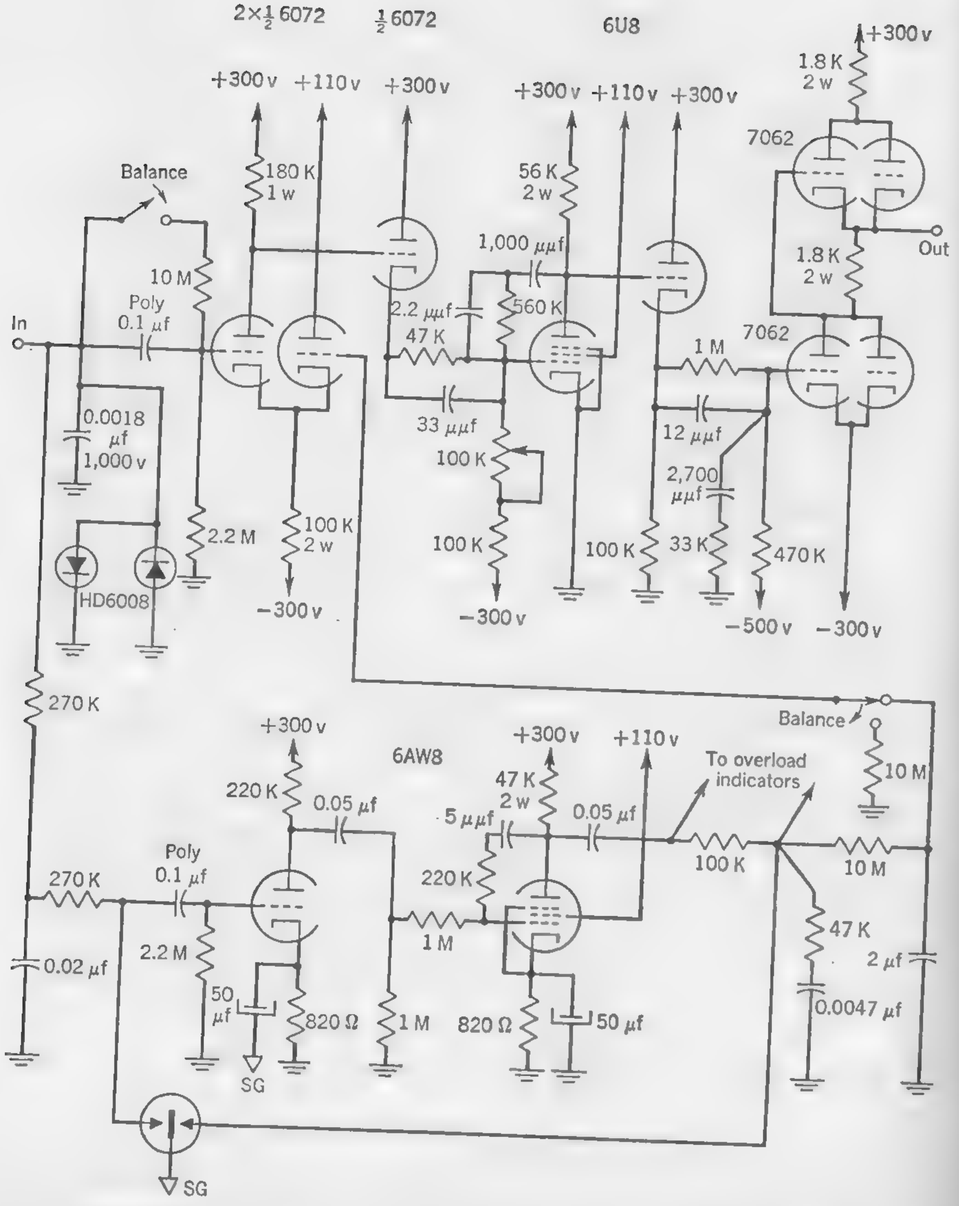


FIG. 2.5.5. Electronic Associates, Inc., Model 6.002 d-c amplifier. 94-cps chopper excitation minimizes hum-pickup effects. Zener diodes prevent large overload voltages on summing point and grid-blocking capacitor. Quadruple unit. Note that the balancing switch disables the chopper-stabilizer channel.

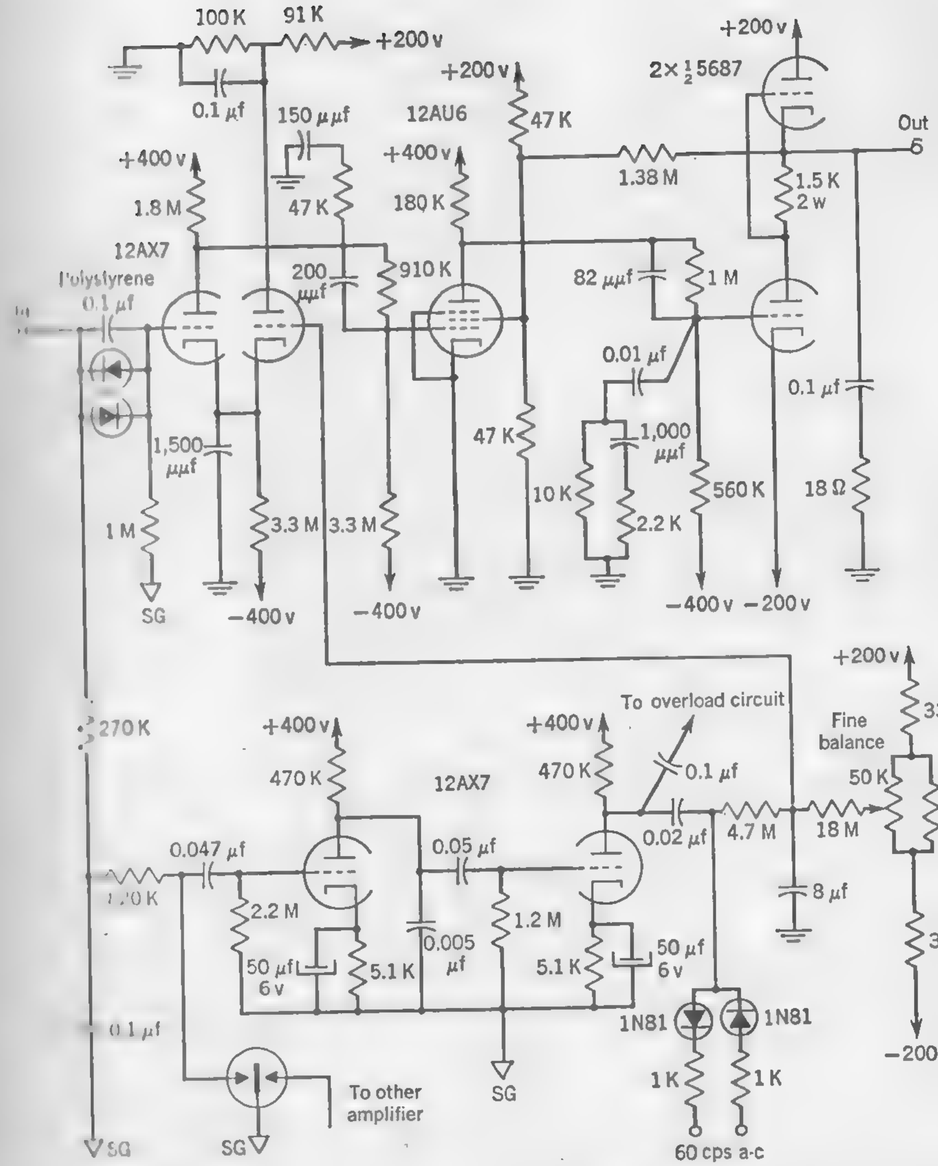


FIG. 2.5.6. Reeves Instrument Corporation Model A-400-1 d-c amplifier. This amplifier employs a grid-blocking capacitor, which is by-passed by TI 622 C diodes to permit fast recovery after overloads.

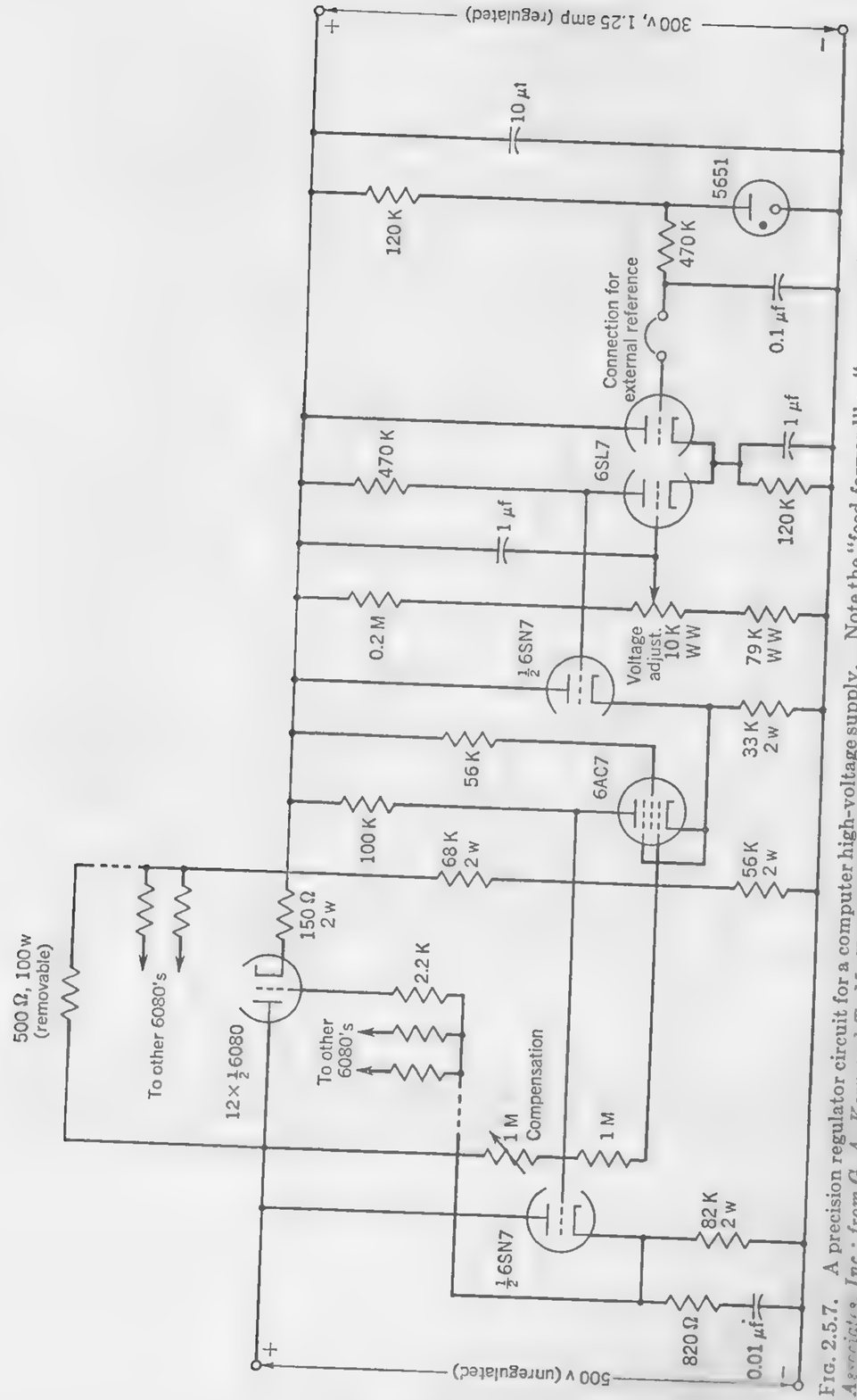


FIG. 2.5.7. A precision regulator circuit for a computer high-voltage supply. Note the "feed-forward" or "compensation" control. (*Electronic Associates, Inc.; from G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

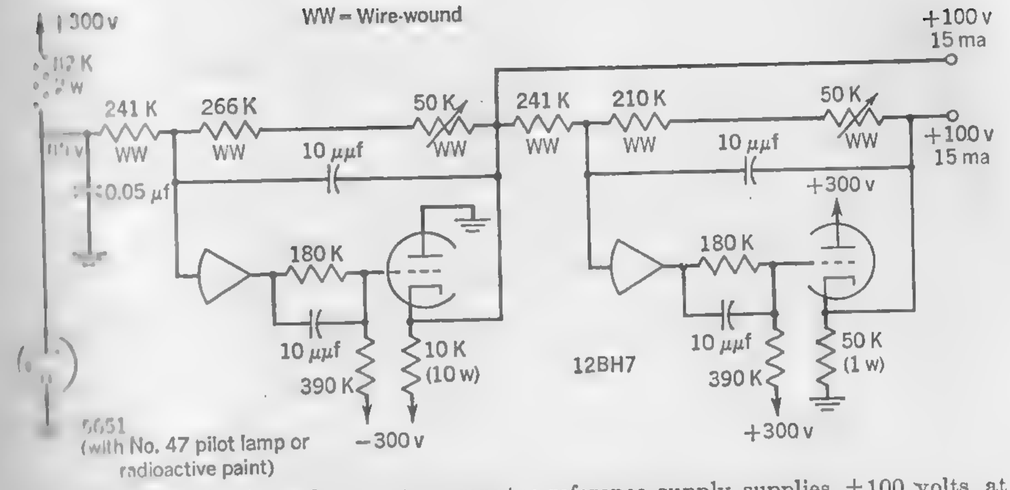


FIG. 2.5.8. This simple and accurate computer reference supply supplies ±100 volts at 15 ma. A VR tube or Zener reference voltage is amplified by a pair of low-cost d-c amplifiers (Philbrick K2-W) to yield output impedances less than 0.1 ohm and regulation within 0.1 volt from no load to full load. Chopper stabilization can be added to reduce the long-term drift to below 0.2 volt. Additional cathode followers can be paralleled for larger loads. If a battery reference is to be substituted for the VR tube shown, a "bootstrap" resistor from the positive output terminal must be connected to the positive reference terminal to supply just enough current for zero battery drain. (*C. Foiles, J. Hartmann, and H. Koerner, Electronic Design, May 10, 1959.*)

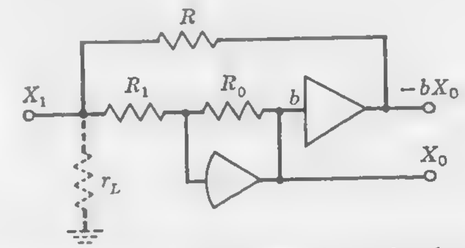


FIG. 2.5.9. This bootstrap circuit supplies all current drawn by the load, and thus eliminates all loading effects, if

$$R = \frac{r_L}{r_L + R_1}(bR_o - R_1)$$

Most frequently, the input resistor R_1 of the operational amplifier is the only load, so that $r_L = R_1$ and $R = bR_o - R_1$. The bootstrap principle has many other applications.

Note that frequency-response data depend radically on the installation (summing-point and load capacitances) and on capacitor compensation of input and output resistors (Sec. 2.4). Similarly, offset and noise specifications can be compromised by ground currents and leakage into summing points (Sec. 4.4).

2.5.2. High-voltage Power Supplies. With chopper- or commutator-stabilized d-c amplifiers, low power-supply impedance (common impedance) is as much a design objective as voltage regulation; both objectives are realized at low frequencies by means of feedback-controlled series regulators and at high frequencies by large by-pass capacitors. The design of accurately regulated high-voltage supplies is treated in the literature;[1,2,3] Figure 2.5.7 shows a typical example. The multiple positive and negative power supplies required for electronic analog computers can be slaved to a common VR tube, Zener, or battery reference by suitable resistance networks.[2]

2.5.3. Reference Power Supplies. The ±100-volt *reference power supplies* which ultimately set the scale of all machine variables in the computer can also be series-regulated power supplies slaved to a common reference. A more accurate and economical scheme, even in small computers, involves precise amplification of a VR tube or Zener reference voltage by ordinary operational d-c amplifiers fitted with special power output stages to supply up to 250 ma (Fig. 2.5.8). Note that such an arrangement does not require extra transformer-rectifier-filter units. To obtain the utmost in accuracy one can similarly amplify a standard-cell voltage. A bootstrap amplifier loop (Fig. 2.5.9) prevents discharge of the standard cell; a relay disconnects the cell whenever the bootstrap circuit is inoperative.

References

1. Greenwood, I. A. Jr., J. V. Holdam, Jr., and D. MacRae, Jr.: *Electronic Instruments*, chap. 15, McGraw-Hill Book Company, Inc., New York, 1948.
2. Korn, G. A., and T. M. Korn: *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.
3. Truxal, J. G.: *Control Engineers' Handbook*, Sec. 11, McGraw-Hill Book Company, Inc., New York, 1958.

Section 3

ELECTRONIC ANALOG COMPUTERS: MULTIPLIERS AND FUNCTION GENERATORS

By

BERNARD D. LOVEMAN, *Project Engineer, Project CYCLONE, Reeves Instrument Corporation, Garden City, N.Y.*

CHARLES D. MORRILL, *Manager, Computer, Guidance, Simulator, and Test Equipment Engineering, Goodyear Aircraft Corporation, Akron, Ohio*

GRANINO A. KORN, *Professor of Electrical Engineering, University of Arizona, Tucson, Ariz.*

CONTENTS

PART 1: COMPUTER SERVOMECHANISMS AND SERVO RESOLVERS

Bernard D. Loveman

Introduction

3.1.1. Computer Servomechanisms. 3–2

Multiplying Servos

3.1.2. Servo Potentiometers..... 3–4
3.1.3. Potentiometer Loading..... 3–7
3.1.4. Servo Amplifiers..... 3–9
3.1.5. Servo Motors..... 3–11
3.1.6. Mechanical Components..... 3–13
3.1.7. Input and Stabilization Circuits..... 3–14
3.1.8. System Performance..... 3–15

Resolver Servos

3.1.9. Introduction..... 3–19
3.1.10. Induction Resolvers..... 3–19
3.1.11. D-C Resolvers..... 3–23
3.1.12. RECTANGULAR and POLAR Modes of Operation.. 3–23
3.1.13. AGC Circuits..... 3–27
3.1.14. Continuous Resolvers and Velocity Servos..... 3–29

Servo Function Generation

3.1.15. Survey..... 3–33

Tapped Linear Potentiometers

3.1.16. Construction..... 3–36
3.1.17. Loading Circuits..... 3–36
3.1.18. Calibration..... 3–36
3.1.19. Practical Considerations..... 3–38
3.1.20. Functions of Two Variables.. 3–39

PART 2: ELECTRONIC MULTIPLIERS AND RELATED TOPICS

Charles D. Morrill

Introduction

3.2.1. Survey..... 3–40

Multipliers Based on Instrumentation of Physical Laws

3.2.2. Multipliers Employing Electrical and Magnetic Fields... 3–40
3.2.3. Miscellaneous Multiplication Schemes..... 3–41

Logarithmic, Quarter-square, and Triangle-integration Multipliers

3.2.4. Logarithmic Multipliers..... 3–41
3.2.5. Quarter-square Multipliers... 3–42

3.2.6. Triangle-integration Multipliers..... 3–42

Modulation Schemes without Feedback Regulation

3.2.7. AM-AM Multiplier..... 3–47
3.2.8. Open-loop AM-FM Multiplier 3–49
3.2.9. Open-loop AM-PM Multiplier 3–49
3.2.10. Open-loop PAM-PWM Multiplier..... 3–50

Multipliers Employing Feedback Regulation of Dual Variable-gain Devices

3.2.11. Feedback Regulation of Gain. 3–51
3.2.12. Time-division Multipliers.... 3–51
3.2.13. Ripple Filtering vs. Frequency Response in Modulation-type Multipliers..... 3–52

Electronic Division and Electronic Resolvers

3.2.14. Electronic Division..... 3–56
3.2.15. Electronic Resolvers and Inverse Resolvers..... 3–56

PART 3: ELECTRONIC FUNCTION GENERATORS, SWITCHING CIRCUITS, AND RANDOM-NOISE GENERATORS

Granino A. Korn

Diode Limiters and Switching Circuits. Operational Relays

3.3.1. Basic Diode Limiter Circuits. 3–62
3.3.2. Precision Limiter Circuits.... 3–62
3.3.3. Comparators, Multivibrators, and Other Switching Circuits. 3–63
3.3.4. Diode Specifications. High-frequency Behavior of Diode Circuits..... 3–63

Diode Function Generators

3.3.5. Special-purpose Function Generators. Use of Smoothing Oscillators..... 3–66
3.3.6. Diode Universal Function Generators..... 3–68
3.3.7. Function Setting and Storage 3–68

Cathode-ray-tube Function Generators

3.3.8. Photoformers..... 3–68
3.3.9. Cathode-ray-tube Scanners... 3–68
3.3.10. Appraisal..... 3–70

Miscellaneous Function Generators

3.3.11. Varistor Function Generators 3–73
3.3.12. Analog-digital Function Generators..... 3–75
3.3.13. Low-frequency Test Oscillators..... 3–75

Functions of Two or More Variables

3.3.14. Use of Conventional Computing Elements. Interpolation Schemes..... 3–78
3.3.15. Use of Diode Function Generators..... 3–79
3.3.16. Cathode-ray Function Generators..... 3–79
3.3.17. Analog-digital Function Generators..... 3–80

Random-noise Generators

3.3.18. Noise-generator Specifications 3–81
3.3.19. Noise Generators Employing Radioactive Sources..... 3–81
3.3.20. Demodulator- and Sampling-type Noise Generators..... 3–81

PART 1: COMPUTER SERVOMECHANISMS AND SERVO RESOLVERS

By Bernard D. Loveman

INTRODUCTION

3.1.1. Computer Servomechanisms. Computer servomechanisms are used for the multiplication of two variables, for the resolution of vectors, and for the generation of functions of one or more variables. An analog-computer servo is an electromechanical feedback system which controls the position or rate of a potentiometer or resolver shaft rotation in response to a computer signal. For servo-design procedures see Refs. 4 to 6 and 19; in this section, performance factors of servo components will be described, and their effect on the system performance will be discussed.

A block diagram of a position servo driving several potentiometers (usually called a multiplying servo) is shown in Fig. 3.1.1. The difference between the input signal X and the follow-up potentiometer voltage is amplified and fed to a control motor coupled mechanically to the follow-up potentiometer and to a multiplying potentiometer. The signals applied to the former are $+E$ and $-E$ volts and to the latter $+Y$ and $-Y$ volts. Stabilization circuits improve the dynamic performance. The output voltage from the multiplying potentiometer is $y = XY/E$, with zero at the center. The voltage E is usually constant and equal to the reference voltage (100 volts). By adding potentiometers to the servo shaft, one can multiply the variable X by other signals to form the products $XU/100$, $-XV/100$, $-XW/100$, etc. The variable X will be

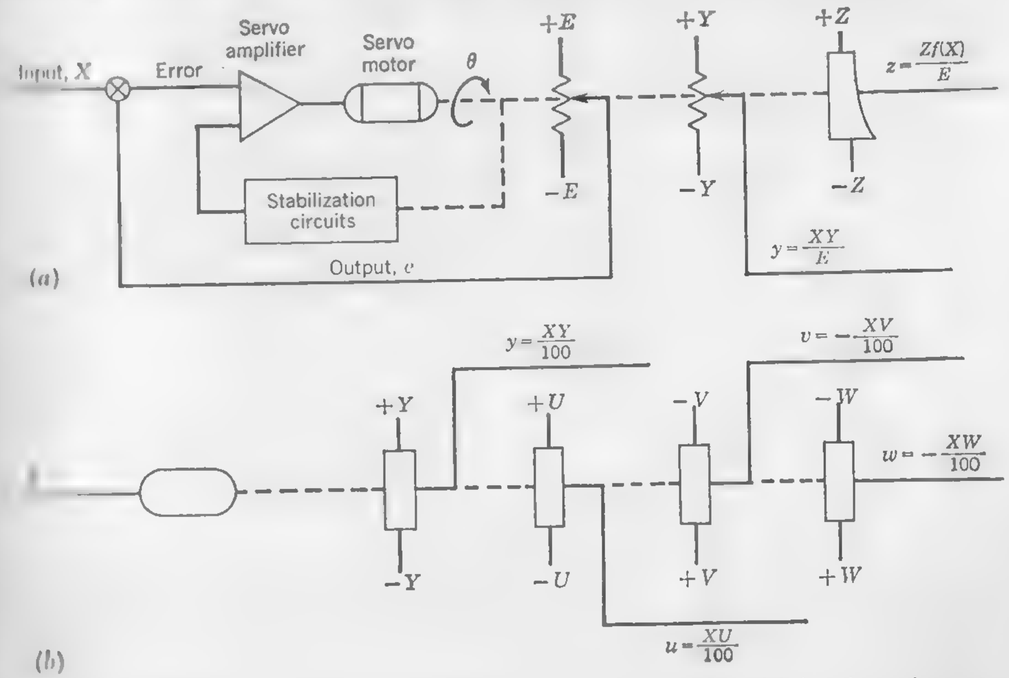


Fig. 3.1.1. Position servo. (*a*) Block diagram. The servo is shown driving three elements: the follow-up potentiometer with $\pm E$ volts across it, a multiplying potentiometer with $\pm Y$ volts, and a function potentiometer with $\pm Z$ volts. The servo system drives the follow-up to reduce the error to zero. Then $y = XY/E$ and $z = Zf(X)/E$. (*b*) Computer operator's symbol for a servo. The follow-up is not shown, and the voltage across it is assumed to be ±100 volts. Each output product may be positive or negative.

called the *master input; Y*, *U*, *V*, and *W* are called the *slave variables*. Since the reference voltage across the follow-up is plus and minus 100 volts, the master variable may be positive or negative (clearly the slave voltages may be of either sign): such a servo is a *four-quadrant multiplier*. In this section, all servos are assumed to have plus and minus 100 volts across the follow-up, unless the contrary is stated.

In the majority of commercial servomultipliers the d-c error signal is modulated and applied to an audio amplifier and a-c servo motor. The principal reason for not using an all-d-c loop has been the drift of d-c amplifiers, which necessitates the use of drift stabilization circuits (Sec. 2.3). These are slightly more expensive than a low-cost reliable chopper used as a modulator; the need for d-c-motor brush and commutator maintenance is also a minor nuisance. The a-c amplifier and servo motor do not cause any special problems. Although this section will deal mainly with a-c servos, all d-c servos have been very satisfactory (Fig. 3.1.9).

The choice of the carrier frequency determines the theoretical upper limit on the servo bandwidth. The loop gain at a frequency of one-half the carrier must be attenuated (see Ref. 19). In actual practice, the bandwidth is a much smaller fraction (typically less than one-tenth) of the carrier frequency. Commercial computer servos

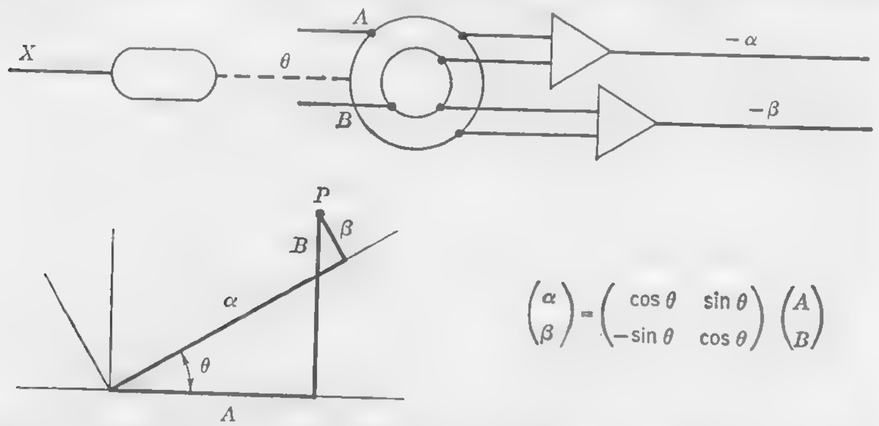
use 60- or 400-cps carriers, for which components (choppers and motors) are available at a reasonable cost; 60-cps a-c is readily available; 400-cps servos have a better frequency response. The need for improved dynamic performance has led to increased use of 400-cps servos.

The advantages of servos as analog multipliers include:

1. Low cost per product.
2. No errors due to the slave variables (the product is directly proportional to the variables Y, U, etc.).
3. No adjustments are normally required.

The principal disadvantage is the dynamic limitation imposed on the master variable. These include limits in servo bandwidth, maximum velocity, and maximum acceleration.

In addition to linear potentiometers the servo may drive other computer elements, e.g., a *resolver*. A resolver is an electromechanical transducer with a mechanical



$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix}=\begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}A\\ B\end{pmatrix}$$

FIG. 3.1.2. D-C resolver servo. The follow-up is not shown. The angle θ is proportional to X. The other inputs are A and B. Although not shown, the voltages $-A$ and $-B$ are also required. The application of a resolver to rotation of rectangular coordinates is shown in the diagram and is also defined in matrix notation. In one system the coordinates of P are A and B; in the other they are α and β.

input θ and two electrical inputs A and B. The resolver outputs are two signals $\alpha = A\cos\theta + B\sin\theta$ and $\beta = -A\sin\theta + B\cos\theta$ (Fig. 3.1.2). A servo which drives a resolver (usually called a resolver servo) is used to perform a coordinate transformation of vectors. One can generate nonlinear functions with a servo by using nonlinear potentiometers (special winding or shaped cards), or by loading multitapped linear potentiometers (Art. 3.1.16). If the applied signal is $+Z$ and $-Z$ volts, then the output voltage z will be a function of two variables Z and X, specifically $z = Zf(X)/100$, as in Fig. 3.1.1.

MULTIPLYING SERVOS

3.1.2. Servo Potentiometers. The performance of a servomultiplier depends in large measure upon the characteristics of its potentiometers. These units are similar in construction to coefficient potentiometers (Sec. 2.1), and characteristics, such as resolution, end resistance, loading, and power rating, apply equally to servo multiplying potentiometers. In addition, several other characteristics are important for servo applications; these include *linearity*, *conformity*, *noise*, and *maximum velocity*.

The accuracy of a potentiometer is expressed in terms of its *linearity error*, i.e., the deviation of the transfer characteristic (output voltage vs. shaft position) from a straight line. The error is expressed in per cent of full-scale voltage applied across the terminals. Every potentiometer has some *end resistance* (resistance of the wires connecting the coil ends to the terminals); this results in three types of linearity specifica-

tions: *independent*, *zero-based*, and *terminal linearity* (Fig. 3.1.3). For computer servos, the third is the preferred method for evaluating and specifying linearity. Although independent or zero-based-linearity potentiometers are slightly less expensive than linearity units for a specified linearity, padding resistors must be added (at

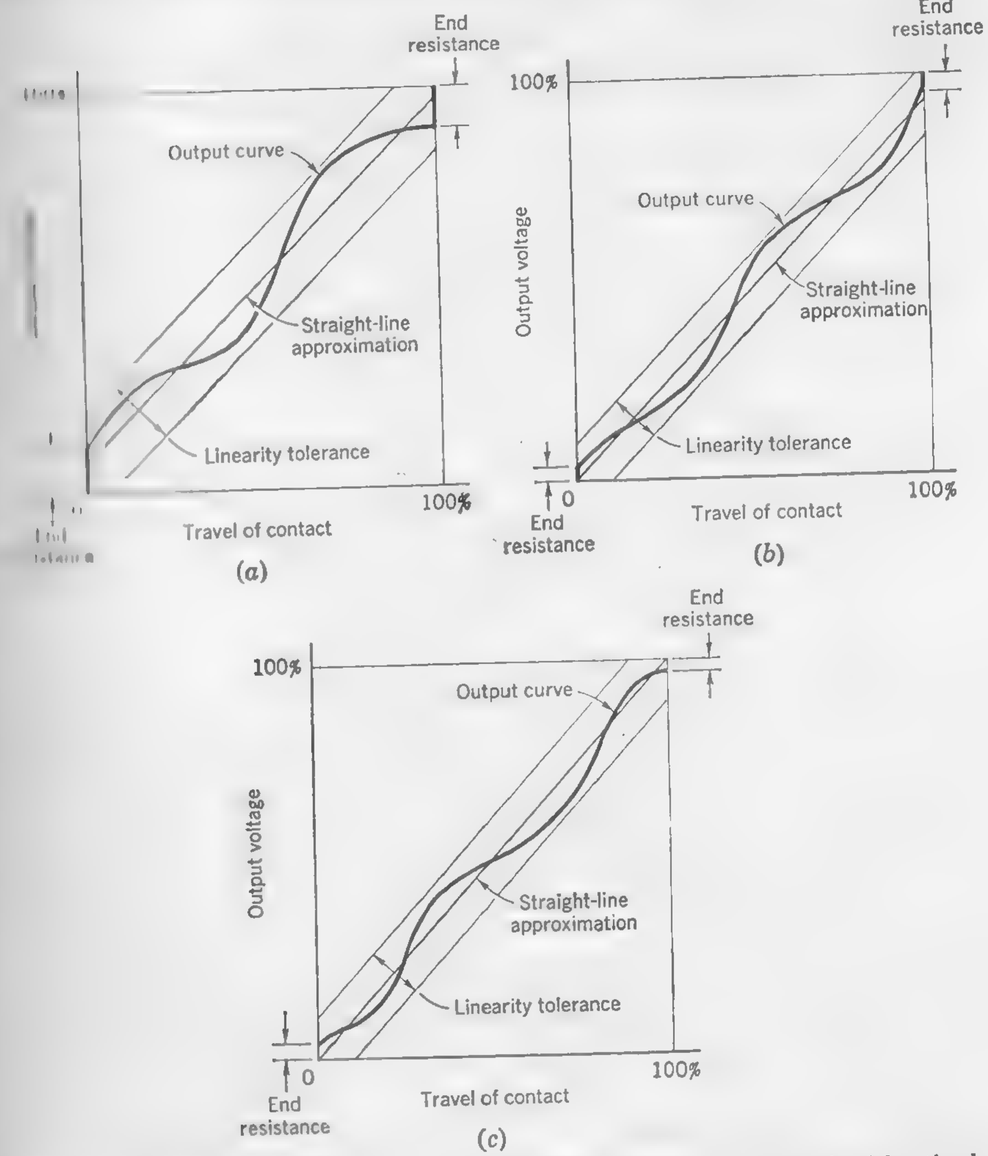


FIG. 3.1.3. Potentiometer linearity: (*a*) *Independent linearity* is determined by the best straight line that can be drawn through the output curve. (*b*) *Zero-based linearity* is determined by the best straight line that can be drawn through the output curve and through the origin. (*c*) *Terminal linearity* is determined by the maximum deviation of the output curve from a theoretically perfect output curve, i.e., a straight line passing through the origin and the 100 per cent point.

extra cost) at the ends of the winding, so that the over-all performance becomes equivalent to the terminal-linearity potentiometer (Fig. 3.1.4). In the following discussion the term *linearity* will be used in the sense of *terminal linearity*.

Since the accuracy of each output product depends upon two potentiometers, a more significant criterion for accuracy is the *conformity* between the follow-up and each

multiplying potentiometer. One measures the conformity error by applying +100 volts and −100 volts across each potentiometer and measuring the output difference between the follow-up and each multiplying potentiometer (see also Sec. 4.4.7). To express the conformity error in per cent, the error voltage is divided by 200. If each potentiometer has a linearity of L per cent, the conformity error will be no larger than $2L$ per cent, provided that each multiplying potentiometer is properly aligned with the follow-up. The one disadvantage of aligning the potentiometers so as to minimize the conformity error is that the electrical zeros may not necessarily coincide; i.e., if the master variable $X = 0$, the output of the Y potentiometer will be zero

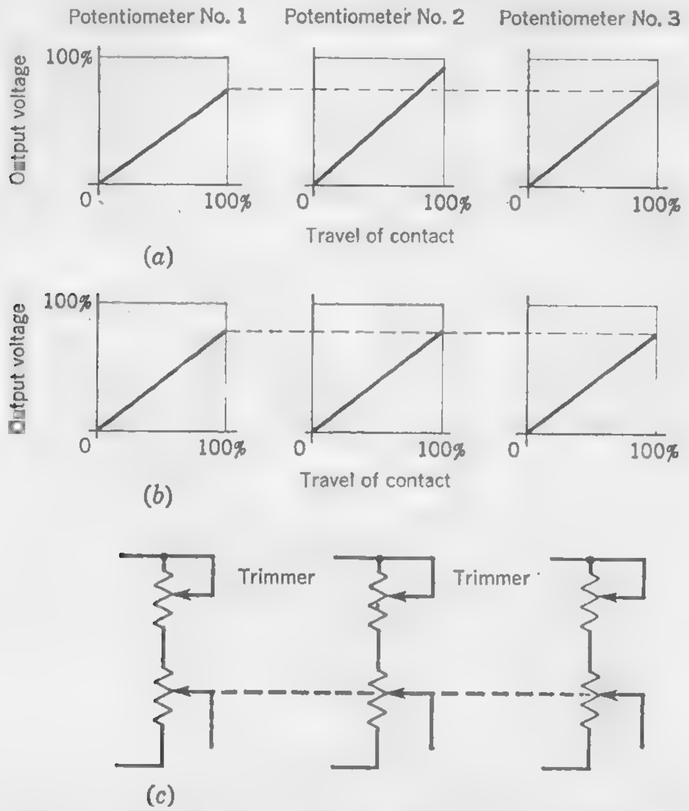


FIG. 3.1.4. Padding to compensate for zero-based linearity. (*a*) Linearity curves before adjustment. (*b*) Linearity curves after adjustment. The trimmer for potentiometer 1 is set to zero because it has the least output at 100 per cent travel. (*c*) Schematic. A similar circuit is used for independent linearity potentiometers except that two trimmers are required for each potentiometer.

$\pm 4YL$ volts. For example, for $Y = 100$ volts and $L = 0.025$ per cent the output (error) may be as large as 100 mv.

If the potentiometers are provided with center taps, the potentiometers can be adjusted so that the taps are aligned. In this case the center taps are grounded in normal use (see Fig. 3.1.5), and if $X = 0$ the output of each multiplying potentiometer is zero. This advantage is obtained at the expense of increased over-all conformity error ($4L$ per cent), and this condition is aggravated by any error in the location of the center taps, typically 1 deg. Consequently the first method of alignment is preferred for general-purpose applications.

The output *electrical noise* in potentiometers arises from various kinds of noise, such as loading-current noise, shorting noise, resolution noise, generated noise, and high-velocity noise (see Ref. 10). The noise in the follow-up potentiometer deteriorates the servo performance; noise generated in the multiplying potentiometers is fed into the

computer circuits as a signal and introduces computing errors. A circuit that has been used extensively for measuring potentiometer noise is shown in Fig. 3.1.6. This test circuit is artificial for computer servo applications; a more realistic and preferred method for evaluating potentiometer noise is that shown in Fig. 4.7.11. In this test

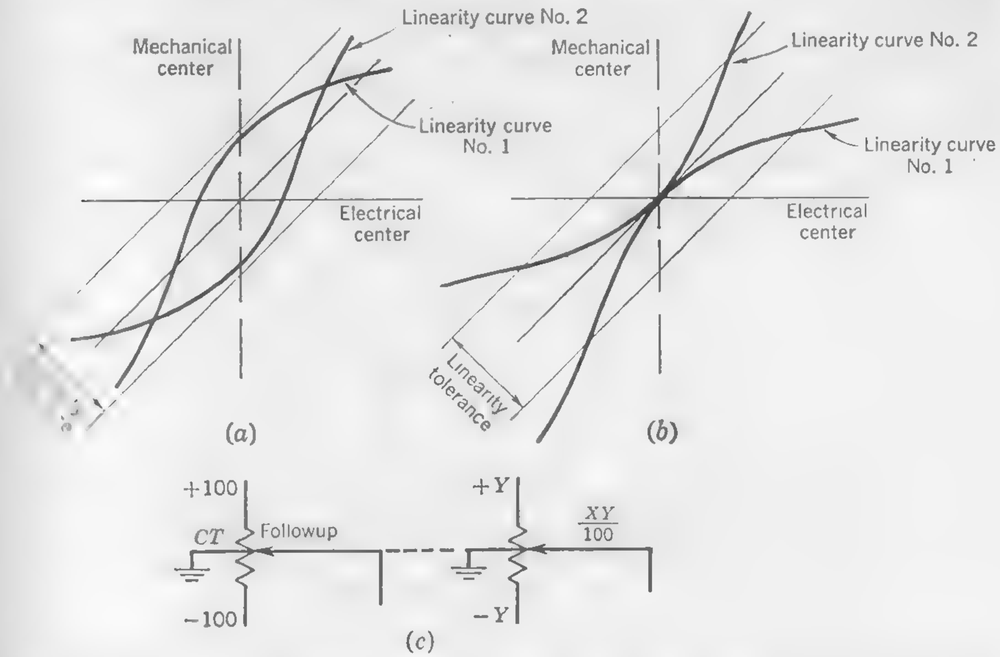


FIG. 3.1.5. Servo potentiometers with grounded center taps. Only the central portion of the linearity curves is shown. (*a*) Linearity curves before grounding the center taps. (*b*) Linearity curves after grounding the center taps. This error may increase if the center taps are not located exactly at the mechanical center. (*c*) Schematic.

the potentiometers are checked in the same circuit used in the multiplier servo loop, and both conformity and noise are evaluated simultaneously. Noise specifications depend upon the potentiometer conformity; typical values are given in Table 3.1.1.

The *maximum velocity* at which a servo potentiometer is driven determines its useful life; usually the first sign of deterioration in a potentiometer is an increase in noise, and the end of its useful life is reached when the noise exceeds twice the specifications. The allowable maximum velocity depends upon the potentiometer diameter; typical values for a useful life of one million cycles are given in Table 3.1.2. This life can be increased by designing special circuits to prevent the servo from running at high speed at those times when an accurate output is not required (see SLO-RESET, Art. 4.1.10).

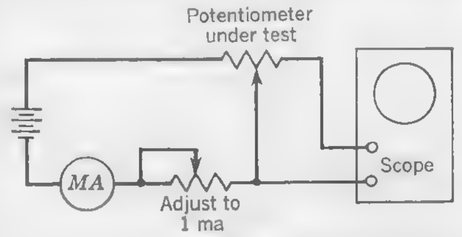


FIG. 3.1.6. Potentiometer noise-test circuit. This method has been widely used. However, the circuit shown in Fig. 4.7.11 is preferred (see text).

3.1.3. Potentiometer Loading. In multiplying servos, each output product is usually connected to an amplifier input. Since this input is a finite resistance, a *loading error* will result (Ref. 2); this error should be less than the linearity error. The magnitude of the loading error depends upon

1. The circuit configuration, e.g., Fig. 3.1.1 or Fig. 3.1.5*c*.
2. The angular position.
3. The ratio of the loading resistance R to the potentiometer resistance r. Let $\rho = R/r$.

For the preferred circuit (Fig. 3.1.1), a convenient rule of thumb is that the maximum loading error in per cent of full scale is $5/\rho$. For commercial computers the potentiometer resistance varies from 20 to 100 kilohms, and the input resistance varies from 100 kilohms to 1 megohm; the ratio R/r consequently varies from 1 to 50. Thus it is essential to reduce the loading error, and fortunately a simple and convenient method is available. It consists of loading the follow-up potentiometer with the same impedance as each multiplying potentiometer. Then the angular position of the follow-up at which the servo nulls (i.e., the input voltage equals the feedback signal) is offset by an amount sufficient to compensate for loading. The one disadvantage of this method is that a linearly calibrated dial will not accurately indicate the input signal. Any residual error is due to differential loading because of resistance toler-

Table 3.1.1. Servo-potentiometer Noise Specifications
See text for definitions

Conformity, %	*Noise Specification Peak-to-peak, mv*
0.5	200
0.2	133
0.1	100
0.05	50
0.02	20

Table 3.1.2. Maximum Velocities for Servo Potentiometers
n is the number of shaft revolutions for full scale, ±100 volts

Diameter, in.	Allowable max velocity	
	rps	Volts per sec
1	10	$2{,}000/n$
2	8	$1{,}600/n$
3	5	$1{,}000/n$

Table 3.1.3. Servo-potentiometer Characteristics

Performance factors	Wire-wound		Film
	Ten-turn	One-turn	One-turn
1. Diameter, in.	1.75	2	1.25
2. Resistance, kilohms	20	50	50
2*a*. Resistance tolerance, %	±2	±2	±10
2*b*. Tap resistance, ohms	5	5	300
3. Conformity, %	±0.05	±0.2	±0.2
4. Resolution, wire turns	12,000	2,100	Essentially infinite, less than 0.005 deg
5. Noise, peak-to-peak mv at 1.5 rpm	50	133	133
6. Max velocity, volts per sec	150	1,500	1,500
7. Power dissipation, watts	5 at 40°C	1 at 40°C	1 at 20°C
8. Breakaway torque, oz-in. per section	2.0	1.7	0.5
9. Moment of inertia, g-cm² per section	14	5	10

ances of r and R. The maximum residual error occurs approximately at an angular position of 58 per cent, or $100/\sqrt{3}$ measured from the center. The equivalent circuit at this position and a plot of the maximum residual error as a function of resistance tolerance are shown in Fig. 3.1.7. The required tolerance on the loading resistance is readily available. On the other hand, a close tolerance on potentiometer *resistance* is not standard. The best commercial tolerance on the resistance of wire-wound potentiometers is 1 per cent and for most film potentiometers 5 per cent. The latter will produce excessive errors for most applications; i.e., the residual loading error of the potentiometer is larger than the linearity error.

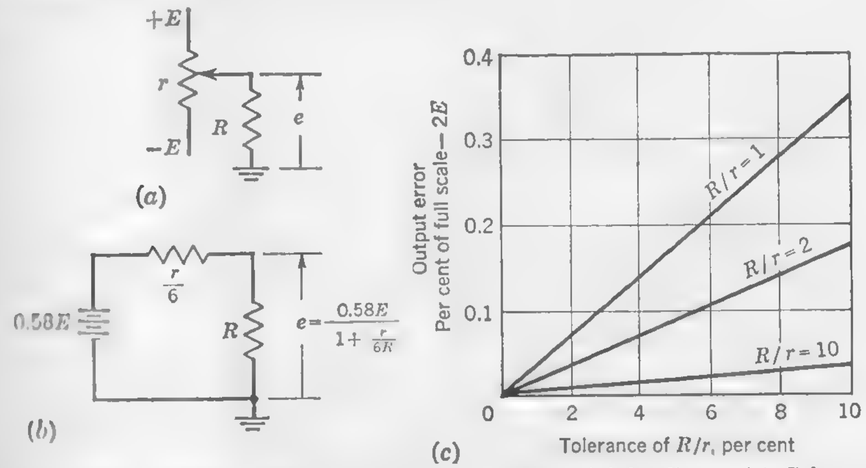


Fig. 3.1.7. Potentiometer residual error due to differential loading. (*a*) Schematic of potentiometer with load. (*b*) Equivalent circuit at the position which produces maximum error due to tolerances in r and R. (*c*) Maximum residual error vs. tolerance of R/r for several values of the ratio R/r.

The use of fuses to protect the potentiometers was discussed in Sec. 2.1. In the case of multiplying servos no error will occur unless the resistance of the fuse in the follow-up differs from that of the fuse in the multiplying potentiometer. Let this difference between the two resistances be denoted by Δ. If $\Delta < LR/100$ (L equals the linearity error in per cent and R is the loading resistance in ohms), then this error will be less than the linearity error.

In commercial analog computers multiplying servos use either one-turn film potentiometers, one-turn wire-wound potentiometers, or ten-turn wire-wound potentiometers. A comparison of these three types is shown in Table 3.1.3. The important mechanical factors (excluding dimensions) are the breakaway torque and the moment of inertia; these are also included in Table 3.1.3.

3.1.4. Servo Amplifiers. In a-c systems the input circuit feeds a phase-sensitive suppressed carrier-modulated signal into the servo amplifier; the latter drives the servo motor. The amplifier can use vacuum tubes, transistors, and/or saturable reactors (magnetic amplifiers). Vacuum-tube circuits are conventional in most commercial analog computer servos; a typical circuit is shown in Fig. 3.1.8. The gain in the operating frequency range is about 66 db, and the output power is 10 watts. A push-pull circuit is used to eliminate the d-c component in the output. To increase efficiency the output stage is tuned. Amplifier output noise will contribute to the amplifier and motor saturation; if it has a quadrature component with respect to the fixed field of the motor (see below), it will produce a spurious torque. In either case, the noise is undesirable and should be kept to a minimum; in particular, with the input grounded the output noise should be less than 10 per cent of the maximum output voltage. In the presence of the inputs the noise, including quadrature and harmonics, may increase to as much as 30 per cent of the output voltage.

A few servos have been constructed with transistors. The major disadvantage, at

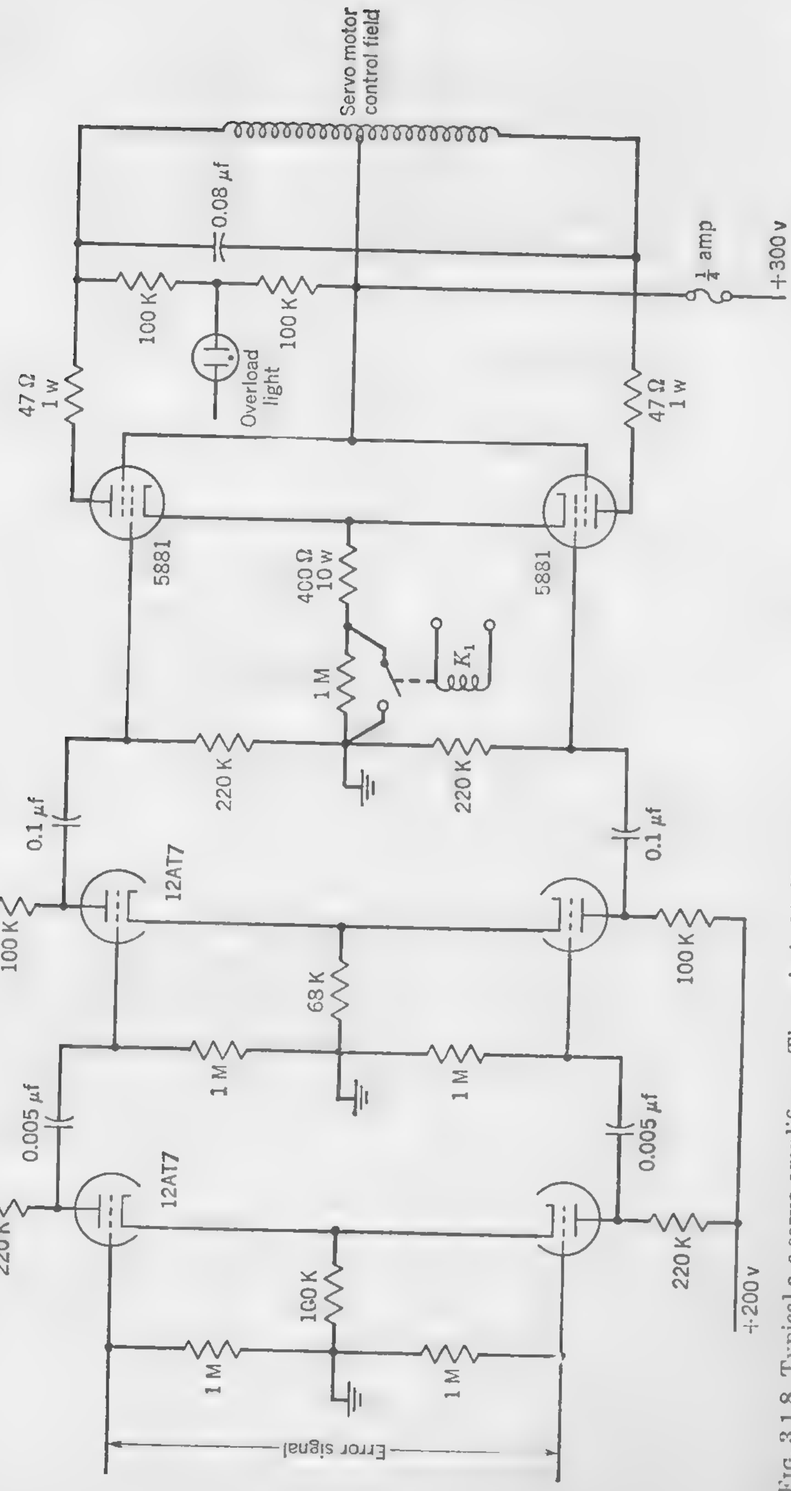


FIG. 3.1.8. Typical a-c servo amplifier. The gain is 66 db and the output power is 10 watts. Note that the rating of the vacuum tubes in the output stage is 20 watts. For operation of overload circuits see Sec. 4.3. K_1 is the ON-OFF relay (*Reeves Instrument Corporation*; see also Sec. 4.1).

this time, is the high cost of transistors in the amplifier and the possible need for separate transistor power supplies.

Compared with vacuum-tube amplifiers, magnetic amplifiers have the advantage of reasonable cost and long life. However, their dynamic characteristics in the linear and the saturated region have been so poor that the dynamic response of the servo system is limited by this factor (Ref. 7). Consequently, these amplifiers have not been used in commercial computer servos.

The servo amplifier for a *d-c system* consists of a drift-stabilized amplifier (Sec. 1.3) with the added requirement that the output stage have sufficient power. A d-c transistor output stage to drive a d-c motor is shown in Fig. 3.1.9.

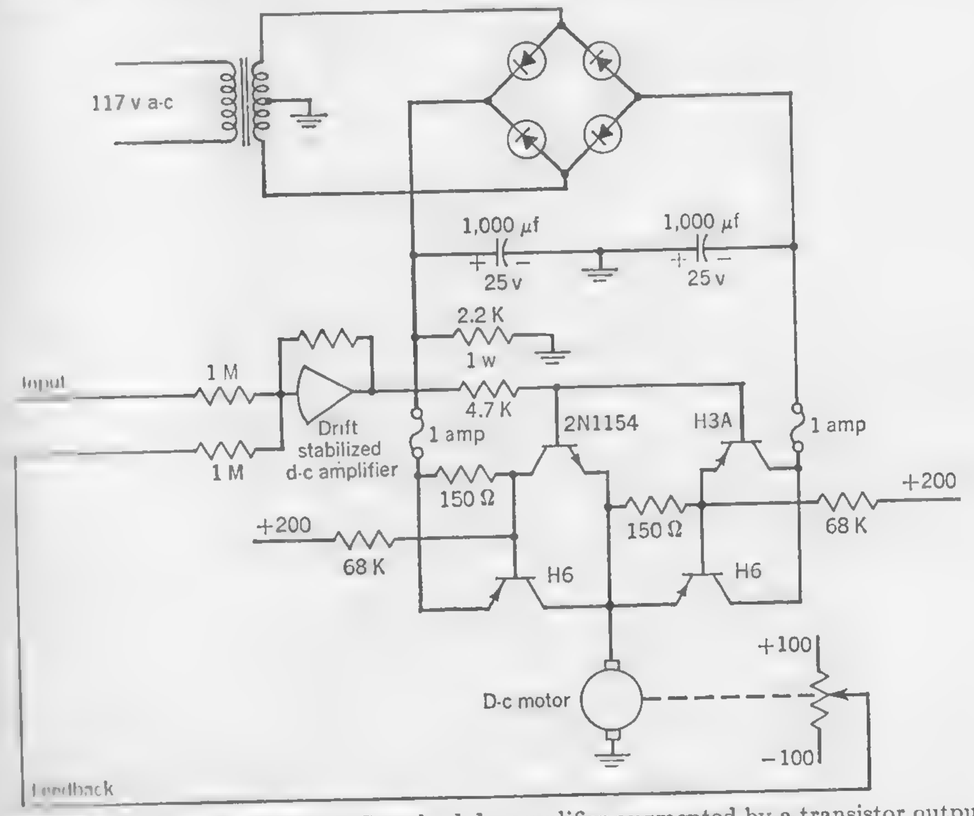


FIG. 3.1.9. D-C servo amplifier. Standard d-c amplifier augmented by a transistor output stage to provide the power necessary to drive the servo motor. The transistors must be protected from excessive output of the d-c amplifier. A series resistor or amplitude limiter may be used (*Goodyear Aircraft Corporation, Akron, Ohio*).

3.1.5. Servo Motors. A-C servo systems are usually driven by a *two-phase induction motor*. One winding is excited by a fixed voltage (*fixed field*) and the second is connected to the output of the servo amplifier (*control field*). The motor develops a torque proportional to the voltage applied to the control field when the two field voltages are 90 deg out of phase. The current due to in-phase voltage is undesirable because it produces no torque and contributes to the saturation of the magnetic field.

The chopper modulator (see below) produces a phase shift of 25 to 65 deg with respect to the excitation, and the additional phase shift necessary to make 90 deg can be developed with an *RC* network. Either the fixed-field voltage or the chopper excitation can be phase-shifted with respect to the line voltage. The first method has the disadvantage that the high impedance of the phase-shift network will usually reduce the motor damping; hence the second method is usually preferred.

The control field is usually available with a low-impedance coil (50 to 500 ohms) or

at extra cost with a high-impedance winding (4,000 to 6,000 ohms); the latter can be driven directly from the output stage of a vacuum-tube amplifier.

The linear performance of the servo motor can be described (Ref. 4) by the transfer function $K_m/s(1 + t_m s)$, where K_m is the motor gain constant, s is the Laplace variable, and t_m is the motor time constant in seconds. The motor gain is relatively unimportant, since the amplifier can be adjusted to yield the desired open-loop gain. The time constant determines the over-all forward open-loop characteristic. In terms of

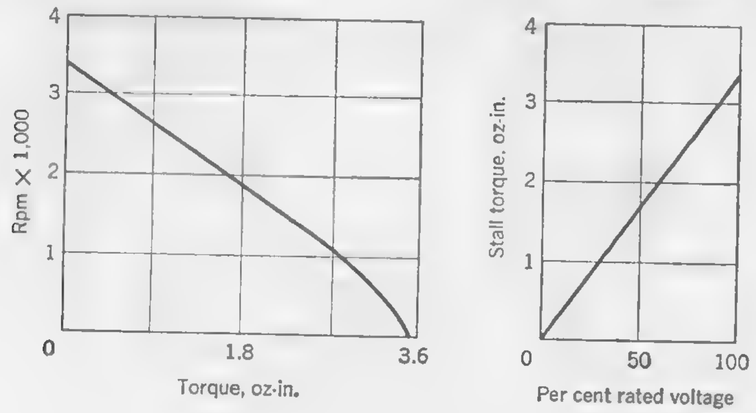


FIG. 3.1.10. Characteristics of two-phase a-c servo motors (see Table 3.1.4). Curves *a* and *b* are performance data for a 60-cps motor and curves *c* and *d* are for a 400-cps motor. For curves *a* and *c* rated voltage is applied to each field. For curves *b* and *d* rated voltage is applied to the fixed field and the control-field voltage is varied.

the Bode plot,* the motor response consists of a 6-db-per-octave slope which breaks into a 12-db-per-octave slope at a corner frequency of $1/t_m$ radians per sec. The motor time constant can be determined from the relation $t_m = M_m J_m$, where M_m is the slope of the speed-torque curve and J_m is the motor moment of inertia. With the motor coupled to its load, the effective inertia will be somewhat greater, and this will reduce the corner frequency. Time constants for high-performance servo motors range between 5 to 50 msec.

Any practical motor will be nonlinear. Characteristic curves for 60- and 400-cps servo motors are shown in Fig. 3.1.10 (see Table 3.1.4). Other important performance factors are:

* Straight-line approximation of amplitude in decibels vs. log frequency.

1. Maximum motor velocity.
2. Maximum motor acceleration.
3. Breakaway voltage—the minimum signal below which the motor will not turn the load. At this voltage the torque becomes equal to the static friction.

D-C computer servo systems are usually driven by a d-c permanent-magnet motor; the output of the servo amplifier is connected to the armature. Performance characteristics (see Fig. 3.1.11) are evaluated in the same way as for a-c motors.

Table 3.1.4. Representative Servo-motor Characteristics
(See also Figs. 3.1.10 and 3.1.11)

Characteristics	Fig. 3.1.11	Fig. 3.1.10*a,b*	Fig. 3.1.10*c,d*
Frequency, cps	d-c	60	400
Max control power, watts	20	8	9
Rotor inertia, g-cm²	4.5	4.0	4.0
Torque at stall, oz-in.	1.7	3.5	2.5
No-load speed, rpm	12,000	3,400	4,800
Time constant, msec	46	6	12
Theoretical acceleration at stall, radians per sec²	27,000	62,000	40,000
Weight, oz	4	12	12

3.1.6. Mechanical Components. Precision mechanical components are required for accurate computer servos. Backlash should be kept to a minimum. If wire-wound potentiometers are used, the maximum backlash of the gear train measured at the follow-up potentiometer shaft should be less than the angular resolution of one-half wire turn. For example, a servo using a ten-turn follow-up potentiometer with 10,000 wire turns satisfies this requirement if it has less than 11 min of backlash. Attempts have been made to reduce backlash by means of anti-backlash gears, but these gears also increase the moment of inertia of the system.

The gear ratio from the motor shaft to the potentiometer arm K_g is a compromise value designed to satisfy the following requirements as closely as possible.

1. The number of steps in the gear train should be as small as possible to minimize backlash. The largest single step in the gear train is limited to 6:1 except in the special case when the motor has a pinion shaft. In this case the first step in the gear train may be as large as 20:1.
2. The maximum motor speed divided by K_g must not exceed the maximum allowable potentiometer speed.
3. The maximum torque at the potentiometer shaft (the maximum motor torque T multiplied by K_g) should be much larger than the friction of the load, say ten times as large.
4. The output acceleration should be as large as possible. The acceleration will be a maximum if the gear ratio is chosen so that the moment of inertia of the motor reflected to the load shaft (that is, $J_m K_g^2$) is equal to the inertia of the load. Under these conditions the maximum acceleration at the load will be $T/2J_m K_g$.

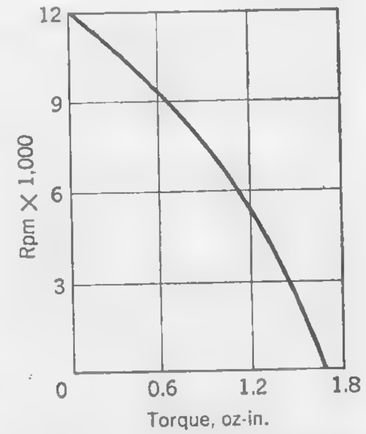


FIG. 3.1.11. D-C servo-motor characteristics. Torque-speed relation (see Table 3.1.4).

The need for limit stops arises from one of the following conditions:

1. The resistance element of single-turn potentiometers covers slightly less than 360 deg; there is a gap between the ends of the element. If the arm of the follow-up travels into this gap the servo will lose the feedback signal and run away.

2. Multiturn potentiometers are usually supplied with mechanical stops, which do not have adequate strength. Additional mechanical-limit stops are used to restrict the travel of the potentiometers to about 98 per cent of the total travel; the master variable is limited to ±98 volts. The stop must withstand the maximum torque exerted by the servo motor multiplied by the gear ratio. In addition, the stop must be able to absorb the kinetic energy of the system, i.e., it should have a low coefficient of restitution. Otherwise the gears may be damaged or slip with respect to their shafts. Gears are usually fastened to replaceable elements, such as potentiometers, by means of friction clamps. A typical lead-screw-type 10-turn limit stop has a torque capacity of 500 oz-in. and can absorb a kinetic energy of 30 in.-oz.

Electrical limit stops are provided in some servos. One method consists of providing taps near each end of the potentiometers. The computer signals are fed to the taps of the multiplying potentiometers (Fig. 3.1.12). The taps of the follow-up are fed with +100 volts and −100 volts; a similar voltage gradient is continued across the end sections of the potentiometer. In this case, the amplitude range of the master variable is ±100 volts. A disadvantage of this method is that full-scale travel may be reduced, and as a result the linearity error will be increased. At some extra cost this difficulty may be avoided by adding extra turns to the follow-up; for example, 11- or 12-turn potentiometers can be used with 10 turns for full scale, ±100 volts.

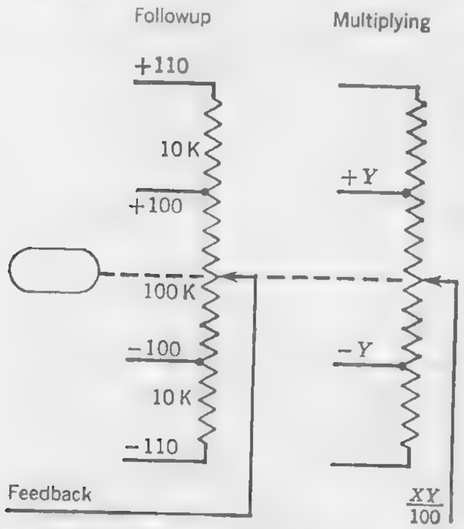


FIG. 3.1.12. Electrical limit stops. The advantage of this scheme is that the full scale of the master variable (servo position) is ±100 volts.

Linearly calibrated *dials* serve two functions. First, they show when the servo is moving and give the approximate magnitude of the master variable. Second, in servos with 10-turn potentiometers the use of two-speed dials allows the servo to be used as an accurate voltmeter. The disadvantages of dials are

1. Added cost (relatively small)
2. Added inertia

Two types of mechanical construction have been used for computer servos. In the first the motor, potentiometers, gear train, and limit stops are fastened to hangers and mounted on a slotted plate. This method provides a large degree of flexibility, and components can be rearranged or special transducers added to fit many requirements. The second method consists of mounting the components in a specially designed casting. The advantages of this method are dimensional stability, greater economy, improved mechanical accuracy, and compactness.

3.1.7. Input and Stabilization Circuits. In a-c servo systems the input circuit generates the d-c error voltage and modulates it to produce a phase-sensitive suppressed-carrier error signal. The essential element of the input circuit is the modulator. A mechanical vibrator or chopper is used in commercial computer servos principally because of its versatility, low noise level, and small unbalance errors. Two basic input circuits are shown in Fig. 3.1.13; each produces a push-pull signal. The *RC* network at the arm of the chopper (1 kilohm and 390 mmf) reduces transients when the chopper contacts break.

The advantage of the first circuit (Fig. 3.1.13*a*) is that no current flows in the input circuit when the servo reaches a null. In other words, the input impedance is infinite. On the other hand, during transients, the current flow in the input circuit may be as large as 600 μa. A servo which has this input circuit may be used as a precision voltmeter, provided its potentiometer linearity error is small, for example, 0.025 per cent. Note that the input and feedback voltage have the same sign at null. In the second

circuit (Fig. 3.1.13*b*) the input and feedback voltages are of opposite sign. The input [illegible] at null is 1 megohm; the maximum input current flow is only 100 μa.

Stabilization circuits are designed to shape the open-loop transfer function to obtain stable, accurate closed-loop response (see also Art. 3.1.8). Two kinds of circuits are used in commercial computer servos, *tachometer feedback* and *network damping*. The former consists of a d-c permanent-magnet generator which is usually mounted directly on or closely geared to, the servo-motor shaft. The network damping circuit consists of an *RC* network connected to the output of the follow-up potentiometer.

Systems with *tachometer feedback* can provide the smoothest (low-error) slow-speed tracking because there is a tighter compensation path around the motor. The disadvantages of tachometer feedback, as compared with network compensation, are

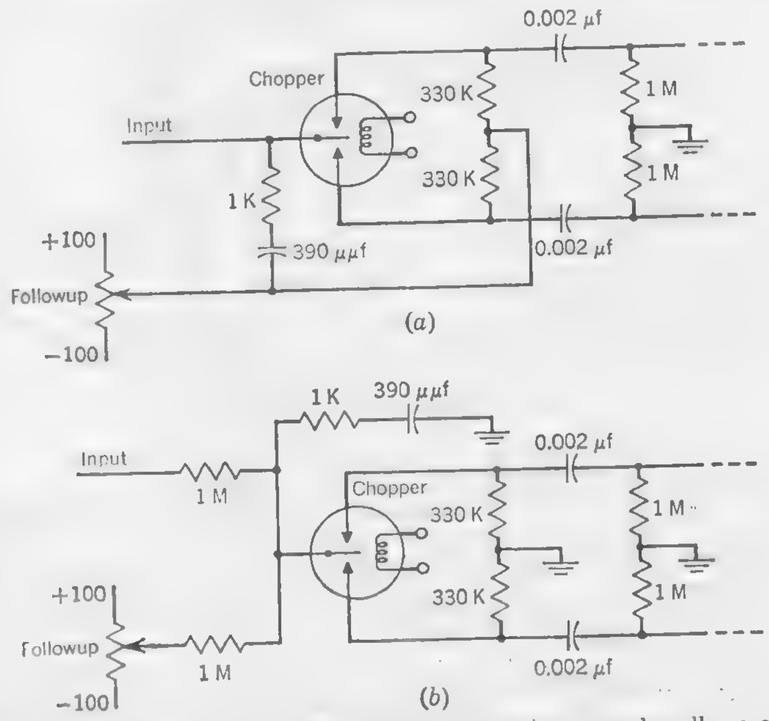


FIG. 3.1.13. Typical input circuits. Each circuit generates a push-pull a-c error signal. The *RC* network at the arm of the chopper (1 kilohm and 390 mmf) is used to reduce transients when the chopper contacts break. (*a*) At null (zero error), the input and feedback voltages are equal, the input impedance is infinite, and there is no loading of the follow-up. (*b*) At null, the input and feedback are equal in magnitude but have opposite sign, the input impedance is 1 megohm, and the loading on the follow-up is 1 megohm.

added cost as well as increased system inertia and friction. With network damping the compensation loop includes the gear train, and the system is relatively ineffective [illegible] intermittent torques such as coulomb friction in the gear train; in addition, [illegible] in the follow-up is intensified by the network.

In both cases simple rate feedback is not adequate to reduce low-frequency dynamic [illegible]. In other words, at low frequencies the open-loop response is characterized by [illegible] (see also Art. 3.1.8). Consequently either a second- or sometimes even a third order system is used in the feedback. The output of the stabilization circuit is summed with the input circuit; typical tachometer and network damping circuits are shown in Fig. 3.1.14.

3.1.8. System Performance. The important closed-loop characteristics of the servo are

1. *Static accuracy*
2. *Tracking accuracy*
3. *Linear response,* including bandwidth, amplitude rise, and dynamic error
4. *Nonlinear response,* including maximum velocity, maximum acceleration, and step response.

Servo-system-design techniques are given in many texts. The purpose of this section is to present test methods for evaluating servo performance and typical numerical values obtained in such tests (Table 3.1.5).

Table 3.1.5. Characteristics of High-performance Computer Servomechanisms
The carrier frequency is 400 cps and the follow-up voltage is ±100 volts
(Reeves Instrument Corporation)

Performance factors	Potentiometers	
	Single-turn	Ten-turn
Static error, mv	50	10
Tracking error, mv	50	25
Servo bandwidth, cps	50	35
Computing bandwidth, cps	2	.8
Resonant peak, db	4	3.5
Max velocity, volts per sec	1,600	140
Max acceleration, volts per sec²	60,000	5,500
Synchronization, msec	100	300

The static accuracy depends upon
1. The motor breakaway voltage v_b in volts
2. The backlash between the motor shaft and the follow-up potentiometer δ in radians
3. The number of shaft revolutions n and the number of wire turns N of the follow-up potentiometer
4. The voltage gradient of the follow-up K_p in volts per radian; for a linear n turn potentiometer $K_p = 100/n\pi$
5. The gain of the servo amplifier K_a

If the first four factors are fixed, the maximum permissible amplifier gain K_A depends upon whether the follow-up is a wire-wound potentiometer (case I), or a film potentiometer (case II).

In case I, $K_A = v_b N/100$, provided $\delta < \pi n/N$.
In case II, $K_A = v_b/K_p\delta$.
Under these conditions the maximum static error ϵ_s in volts is
Case I, $\epsilon_s = 100/N$.
Case II, $\epsilon_s = K_p\delta$.

The above formulas are derived from a consideration of the smallest change in the motor position which will produce a recognizable increment in the follow-up voltage. This incremental follow-up voltage multiplied by the servo-amplifier gain must be no greater than the servo-motor break-away voltage. In case I, the incremental voltage is determined by the follow-up resolution and in case II by the backlash. If the amplifier gain is larger than K_A the servo will chatter (hunt between wire turns or in the backlash) and wear out the potentiometers; if the gain is smaller the static error will increase. In case I if $\delta > \pi n/N$ the servo-amplifier gain must be reduced. The static error is measured by taking the difference between any constant servo input and the corresponding output voltage of the follow-up. As a rule the static error of a high-performance servo will be much smaller (one-tenth) than the conformity error, and then the static-product accuracy depends only upon the conformity error.

One of the most important criteria for computer servo performance is *low-speed*

[illegible]curacy. This criterion is a measure of the ability of the servo to follow [illegible] inputs accurately. The tracking accuracy is measured by driving the [illegible] from an integrator with a constant input voltage. The error in tracking is the [illegible]ence between the integrator output and the follow-up output. A high-performance servo will have a tracking error less than one-half the linearity error with the integrator output changing as slowly as 10 mv per sec; for example, if the follow-up has

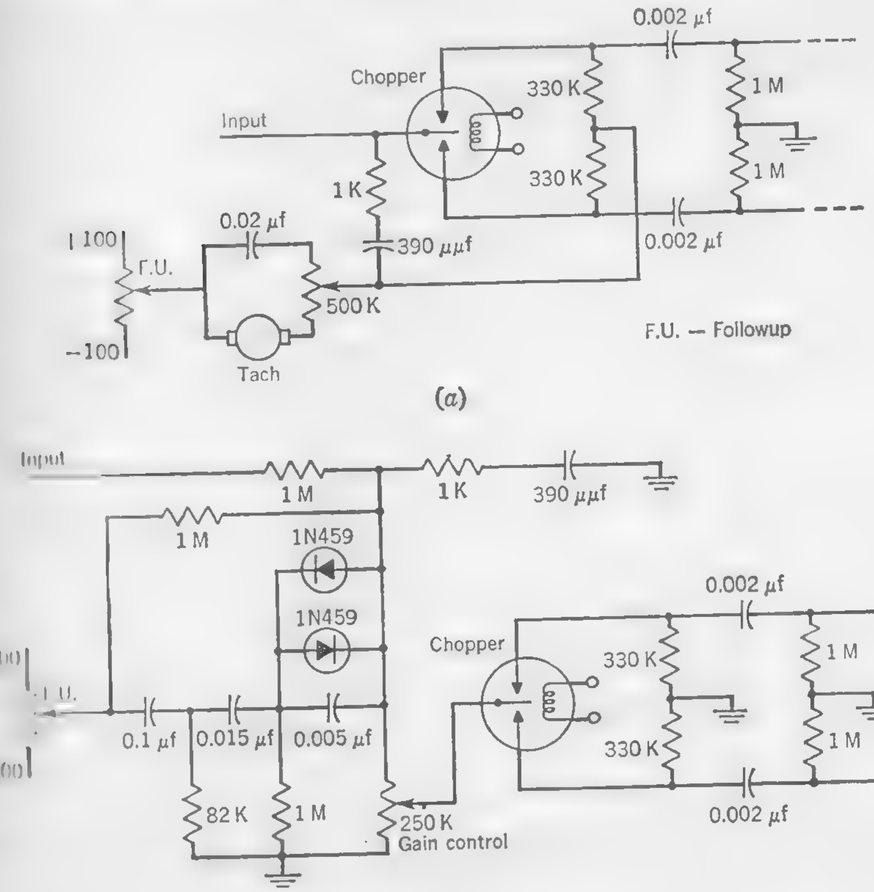


Fig. 3.1.14. Stabilization circuits. (*a*) Tachometer feedback with a high-pass filter to increase the velocity constant (reduce velocity error). (*b*) Network damping. A third-order network is used to improve performance at the computing frequencies. The diodes (1N459) modify the network to provide stability for large error signals, i.e., at low amplifier gain. (*Reeves Instrument Corporation.*)

linearity error of 0.025 per cent, and with a full scale of ±100 volts, this requirement is satisfied by a tracking error less than 25 mv. With a small low-speed tracking error, the product accuracy is not degraded by servo error at low speeds.

The *linear performance* of the computer servo may be completely characterized by its closed-loop frequency response, which is in turn determined by its open-loop characteristics. The inherent forward-loop transfer function is $K_a K_m K_g K_p/s(1 + \tau s)$ where the gain constants K_a, K_m, K_g, and K_p have been defined previously and τ is the system time constant (the servo-motor time constant increased to include the effect of the load). Several synthesis procedures for shaping this characteristic to obtain good closed-loop performance are available. Two typical shaping networks are shown in Fig. 3.1.14.

The closed-loop response may be specified by the *servo amplitude and phase charac-*

teristics. Two properties of special interest can be obtained immediately from the amplitude curve; they are

1. The *bandwidth*, the frequency at which the amplitude response is down 3 db
2. The *resonant peak*, the maximum amplitude of the response

Test circuits for measuring these quantities are simple and well known. A peak in the amplitude response is desirable for a snappy transient response, but this peak should not be too high, in order to preserve system stability; typically the resonant peak lies between 3 and 6 db.

A useful criterion for the computing accuracy of a servo is the *dynamic error*, i.e., the amplitude of the difference between the input and output voltages for a sine-wave input. In other words, the dynamic error is the magnitude of the vector difference

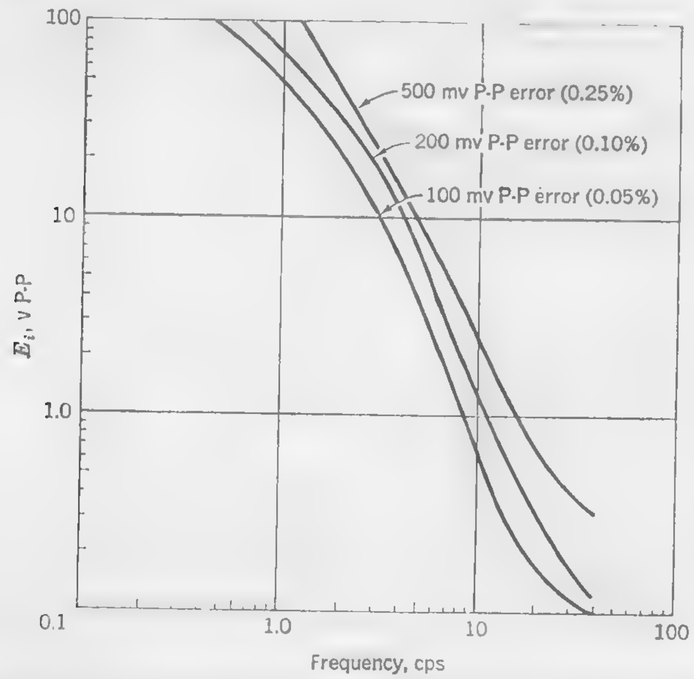


FIG. 3.1.15. Dynamic error response for a high-performance 400-cps servo. Note that the input signal and the dynamic error are expressed in the same units, peak-to-peak voltage. The percentage figures express the error voltages as a fraction of the full-scale follow-up voltage ±100 volts.

between the input and output; it includes both the amplitude and phase components of the difference. The dynamic error depends on the input signal amplitude and frequency. A convenient method for presenting these data is to plot the input signal amplitude vs. frequency for a fixed error; typical curves are shown in Fig. 3.1.15. Contours for several values of dynamic error are shown because of the system nonlinearities. For inputs which are large compared with the dynamic error, the input and output are approximately equal. Hence the ratio of the input to the dynamic error is equal to the magnitude of the open-loop gain. For example, in Fig. 3.1.15 at 2 cps the error with 20-volt peak-to-peak input is 100 mv; consequently the gain at this frequency is 2,000.

The computing frequencies are usually only a small fraction, less than one-tenth, of the servo bandwidth for two reasons. First, at the bandwidth frequency the dynamic error is large compared with the input signal (Fig. 3.1.15). Second, the amplitude of the input must be kept extremely small because of velocity and acceleration limits (see below). A convenient term for evaluating servo dynamic performance is the so-called computing bandwidth, the frequency at which a peak-to-peak amplitude of

0.1 volts produces an error of 0.5 per cent of the input (100 mv); typical values are given in Table 3.1.5.

The two principal *nonlinear performance factors* are the *maximum velocity* and the *maximum acceleration.* The former can be measured in three different ways. In the simplest method a large step input is applied, the resulting output of the follow-up potentiometer is recorded, and its maximum slope is measured. This maximum slope gives the maximum velocity. In a second method a sine-wave input is applied. In the linear range the servo output is also sinusoidal, say $r = A \sin \omega t$. The maximum output velocity is $A\omega$ and the maximum acceleration $A\omega^2$. The amplitude and frequency may be selected in such a way that the maximum acceleration remains small and the maximum velocity becomes large (A large and ω small). The maximum velocity may then be increased gradually until the output becomes distorted because of velocity limiting. This condition may be recognized by the distortion at the zeros of the output (triangular wave shape). The third method for measuring the maximum velocity consists of applying a constant-velocity input (ramp) and measuring the servo error after the initial transient. At the maximum velocity there will be a sharp increase in the error.

The maximum acceleration is more difficult to measure; two methods may be used. The first again uses a sine-wave input, but now the amplitude and frequency are chosen so that the output is within the velocity limit (small A and large ω) and the maximum acceleration is increased until the sine-wave output limits. Acceleration limiting is evidenced as a flattening of the peaks of the output. The maximum acceleration may also be determined by applying a constant-acceleration input (parabola) and measuring the servo error for different values of this acceleration. In this case the error will be small until the maximum acceleration is reached. Here it should be noted that during each run the velocity increases linearly with time and the error increases abruptly when the velocity limit is attained, but unless the acceleration limit is exceeded this happens a finite time after starting.

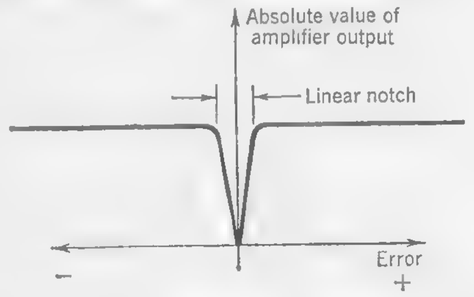


FIG. 3.1.16. Amplitude response curve. The servo is said to be in synchronism when the error lies within the linear notch.

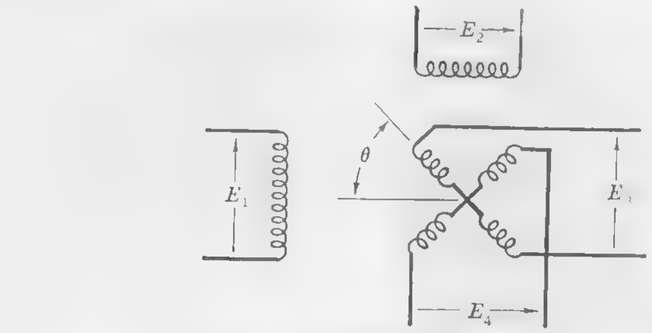
A high-gain servo with good linear transient characteristics can still become unstable for a large step input. This behavior can be explained in terms of another nonlinear factor called the *synchronization.* A servo is said to be in synchronism whenever it is following the input signal with an error small enough to lie in the linear portion of the amplitude-response curve, i e., in the linear notch in the curve shown in Fig. 3.1.16. When a step input is applied, the error will exceed this value temporarily, and the servo saturates. The time required for the servo to return to synchronism is called the synchronizing time. For practical purposes the synchronizing time of computer servos may be defined as the time required for the output to achieve and to remain within 10 per cent of the final value, when the servo is subjected to a 50-volt step. In order to follow large step inputs with good synchronization (without many overshoots) it may be necessary to use nonlinear elements or two-speed circuits (Ref. 6).

RESOLVER SERVOS

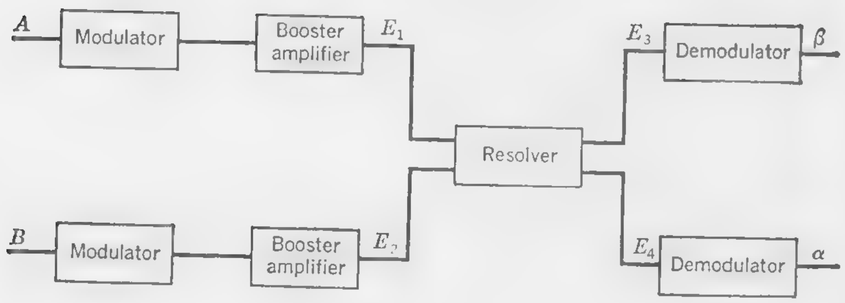
3.1.9. Introduction. A *resolver*, as a d-c analog-computer element, was defined above (Fig. 3.1.2). The accuracy of a resolver is customarily described by its conformity. Conformity is defined as the maximum functional error with one electrical input zero (e.g., $A = 0$). It is usually expressed as a per cent of the full-scale (peak-to-peak) output, i.e., $2B$ volts. Thus a 0.1 per cent conformity resolver means that with $A = 0$ and $B = 100$ volts the errors in the α and β outputs are less than 0.2 volt.

3.1.10. Induction Resolvers. An *induction resolver* is an a-c electromagnetic transducer for performing trigonometric computations.[18] Essentially it is a trans-

former with variable coupling between the primary (stator) and the secondary (rotor) windings. Commercial resolvers are provided with two input stator windings and two output rotor windings (see Fig. 3.1.17*a*). The internal construction of the resolver resembles a two-phase two-pole wound-rotor induction motor with slip rings. The pair of stator windings are at right angles to each other (in space quadrature); the rotor windings are also in space quadrature. The windings are distributed in the slots so that the output voltages are sinusoidal functions of the rotor angular position.



(*a*) $E_3 = E_2 \cos\theta - E_1 \sin\theta$
$E_4 = E_1 \cos\theta + E_2 \sin\theta$



(*b*)

FIG. 3.1.17. Induction resolver. (*a*) Schematic. The a-c input voltages are E_1 and E_2; the a-c output voltages are E_3 and E_4. (*b*) Block diagram. The modulators convert the d-c input voltages to a-c signals. In practice, the resolver does not perform in an ideal manner and compensator circuits are required. These include two auxiliary resolver windings (not shown) similar to the stator windings, and booster amplifiers for isolation. Each booster is a feedback amplifier with a net loop gain of 60 to 80 db. The auxiliary windings provide the feedback voltages for the booster. These circuits improve the linearity of the resolver amplitude characteristic, and they also compensate for variations due to frequency and temperature. The a-c resolver outputs are converted to d-c in the demodulators.

If a-c voltages are applied to the stator windings, the stator currents produce a magnetic flux such that the magnitude of the flux vector is proportional to the vector sum of the stator currents, and the angular position of the vector is equal to the arctangent of the ratio of the two currents. The rotor outputs are then proportional to the magnitude of the flux vector and to the sine or cosine of the angle of rotation of the rotor with respect to the flux vector. Explicitly, the outputs E_3 and E_4 expressed in terms of the inputs E_1 and E_2 and the rotor position θ are the resolver relations:

$$E_3 = E_2 \cos\theta - E_1 \sin\theta$$
$$E_4 = E_1 \cos\theta + E_2 \sin\theta$$

These relations are the same as the equations in Fig. 3.1.2.

The major advantages of an induction resolver are:

1. Small size with high accuracy; for example, a unit 2 in. in diameter (Reeves Instrument Corp. Model R 600), has an output functional error of less than 0.05 per cent of the input voltage.
2. Smooth output (no granularity) as a function of angular position.
3. Negligible wear from use, because electrical contact to the rotor is made through the rings.

The disadvantage of an induction resolver is the requirement for modulators and demodulators (Fig. 3.1.17*b*). If standard techniques are used for modulation and detection, it is difficult to obtain accuracies comparable to that of the resolver itself.

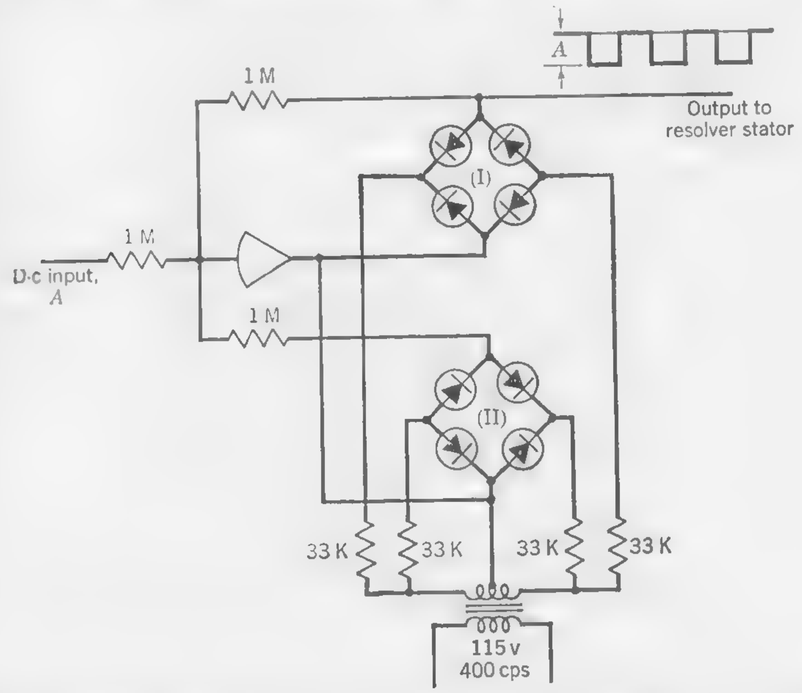


FIG. 3.1.18. Precision gated-amplifier modulator. Diode bridge I conducts on alternate half cycles; this switching provides the required modulation. Bridge circuit II is used to prevent the amplifier from overloading when bridge I is open. The output is a square wave with peak-to-peak amplitude equal to the input voltage.

One special technique for precision modulation is the use of a gated amplifier[8] to control the a-c output amplitude from the d-c input. A typical circuit is shown in Fig. 3.1.18; its output is a 400-cps square wave whose amplitude is strictly proportional to the d-c input. The switching is controlled by the diode-bridge circuits and the output amplitude is determined by the ratio of the feedback to input resistors. In the circuit shown the peak-to-peak value of the output is equal to the d-c input signal. Two modulators are required for each resolver, one for each of the stator inputs A and B.

Resolvers are usually designed for carrier frequencies of 60 cps or 400 cps. They are essentially reactive elements; a typical 400-cps resolver has a bandwidth from 20 cps to 40 kc with a resonant peak at 30 kc. Consequently, if a square wave is applied to a resolver stator, each rotor output will be a square wave with transients at the leading edges. A peak-to-peak detector which suppresses these transients is shown in Fig. 3.1.19. A peak-to-peak circuit is used to prevent errors due to asymmetry in the rotor output. In addition, a zero-to-peak detector would give only one-half the output that this circuit delivers. The detector eliminates output transients by sampling during the center portion of each half cycle of the square wave; its basic components are three choppers and a d-c amplifier. The chopper driving voltages

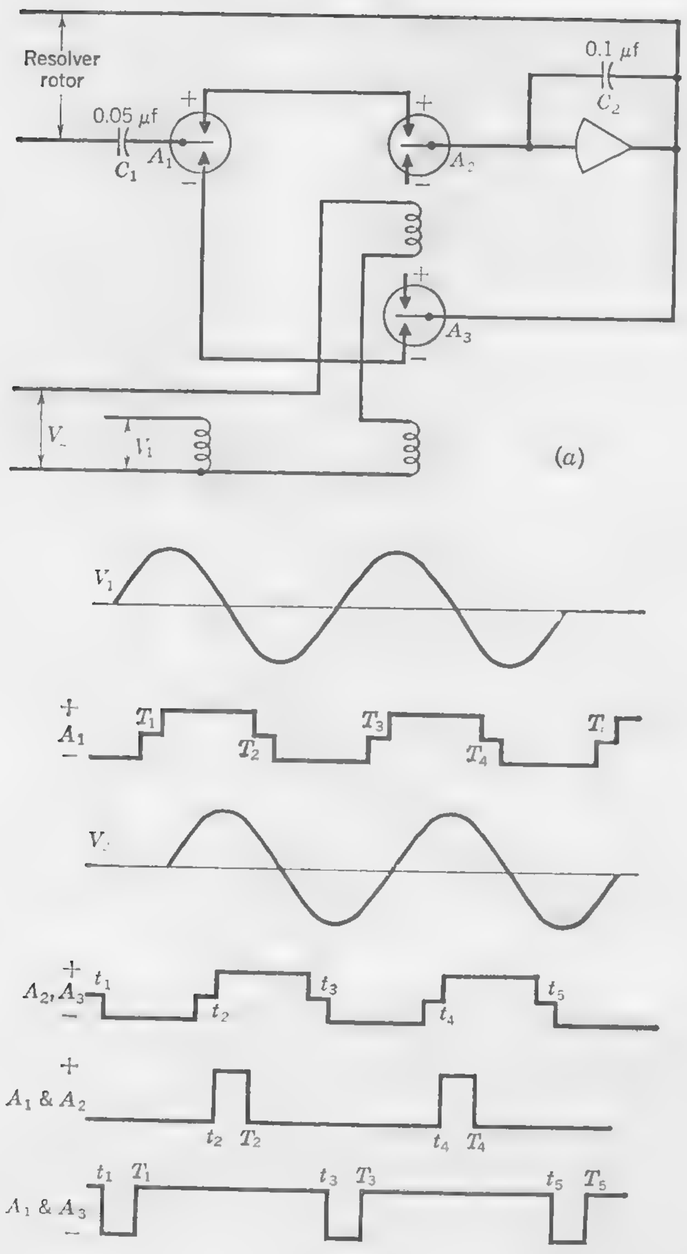


FIG. 3.1.19. Precision peak-to-peak demodulator. (*a*) Schematic. When chopper arms A_1 and A_3 are both connected to their negative terminals, the resolver rotor terminals are connected across condenser C_1 (0.05 μf) and it charges to the peak value of the resolver output voltage. When chopper arms A_1 and A_2 are both connected to their positive terminals, the condenser C_2 (0.1 μf) is charged to the peak-to-peak resolver output, i.e., the peak resolver output plus the voltage stored on condenser C_1. During the time intervals when the resolver circuit is not connected to C_2, the amplifier holds the output constant. (*b*) Timing diagram. Both A_1 and A_3 are negative during the time intervals t_1 to T_1, t_3 to T_3, and t_5 to T_5. Both A_1 and A_2 are positive during the time intervals t_2 to T_2 and t_4 to T_4. The resolver output (not shown) leads V_2 by approximately 20 deg.

is phased so that the resolver output is sampled at the proper time intervals. Another important property of the circuit is its low output impedance, which makes it suitable for driving other computer circuits. Two detectors are required for each resolver, one for each rotor output.

The accuracy of the modulator and demodulator is 0.05 per cent, i.e., the gain from the d-c input to the d-c output is 1 ± 0.05 per cent. The major limitation of this resolution resolver system is that the dynamic performance with respect to the A and B inputs is limited to a few cycles. For example, at 10 cps the total error is 0.5 per cent.

3.1.11. D-C Resolvers. Commercial *d-c resolvers* consist of two or more ganged and carefully aligned sine-cosine potentiometers. D-C computing amplifiers are required to invert the inputs A and B to provide $-A$ and $-B$; amplifiers are also needed to sum the potentiometer outputs to form α and β (Fig. 3.1.2). Three types of *precision wound sine-cosine potentiometers* are used in commercial d-c analog computers.

The first consists of a closely wound rectangular resistance card rotated by the input shaft; four contacts are attached to the case (Fig. 3.1.20*a* and Ref. 9). A typical unit four inches in diameter has a conformity accuracy of 0.5 per cent (e.g., Gamewell Co. Model RL-14MS).

The other two types are rotary potentiometers with the resistance element wound around a core formed into a circle. The voltages ($+A$ and $-A$) are applied to the ends of the coil and the output is obtained from a contact which is fastened to the input shaft. One unit of this kind uses a combination of two methods to provide the accuracy required for the trigonometric functions. The potentiometer consists of a wire-wound mandrel with variable pitch. In addition, loading resistors are connected across taps located on individual turns of the coil (Fig. 3.1.20*b*). A typical unit 6 in. in diameter has a conformity accuracy of 0.14 per cent (Helipot Corp., Model SNX2031). In another method, a shaped card is wound so that the length of wire for individual turns varies and the rate of change of output varies accordingly (Fig. 3.1.20*c*). A typical unit 15 in. in diameter has a conformity of 0.07 per cent (Electronic Associates Inc., Model 315).

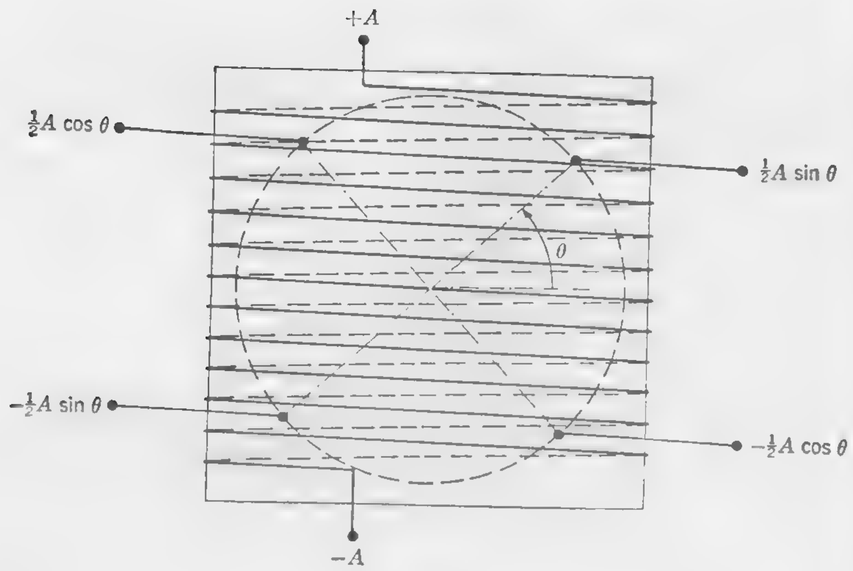
The advantages of the sine-cosine potentiometer are that it is inherently suitable for a d-c analog computer, its simplicity and its relatively, good dynamic characteristics. On the negative side are brush contact noise, limited resolution, lower maximum operating speed, and shorter life; also, for a reasonable size the conformity is relatively poor.

Film potentiometers with sine and cosine outputs have also been constructed. Commercial units are available in which the functional relation is obtained by varying the width of the film element. A typical unit two inches in diameter has a conformity of 0.3 per cent (Computer Instruments Corp., Model 206). A special resolver that has been constructed consists of a cylindrical film resistance element. Electrical contact is made at one end by a set of brushes mounted on the input shaft and at the other end by a segmented metallic ring (Fig. 3.1.21 and Ref. 14). Other methods for generating sine functions are a cam-driven linear potentiometer, or the use of a special mechanical linkage such as a Scotch yoke.

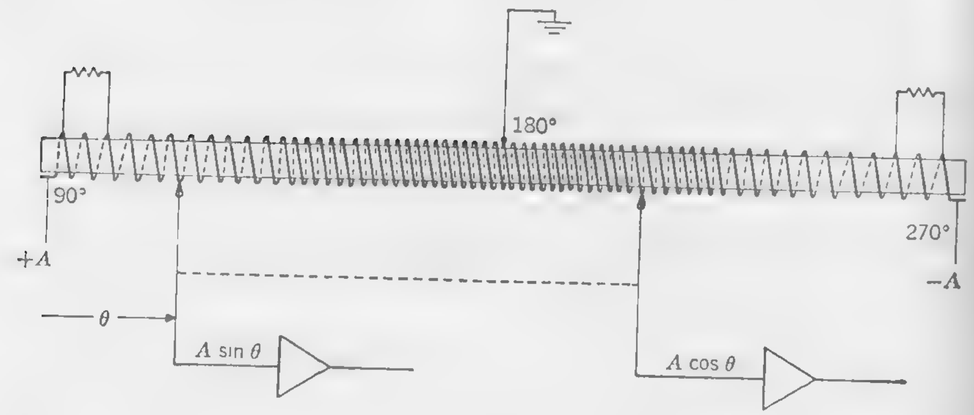
Loading effects must be considered for all of these sine-cosine potentiometers. Their impedance is usually small, 20 to 50 kilohms, and a low-impedance source (such as a d-c amplifier) is required for both the $+A$ and $-A$ inputs. Moreover, the usual values of impedance at the output will cause a loading effect and increase the functional error. In most commercial units the no-load functional relation is modified (*preloading*) in order to take account of this condition and to give the correct result with, say, a 1-megohm load.

3.1.12. RECTANGULAR and POLAR Modes of Operation. In the following sections no distinction will be made between the two basic types of resolvers, induction or d-c, except when it is specifically mentioned.

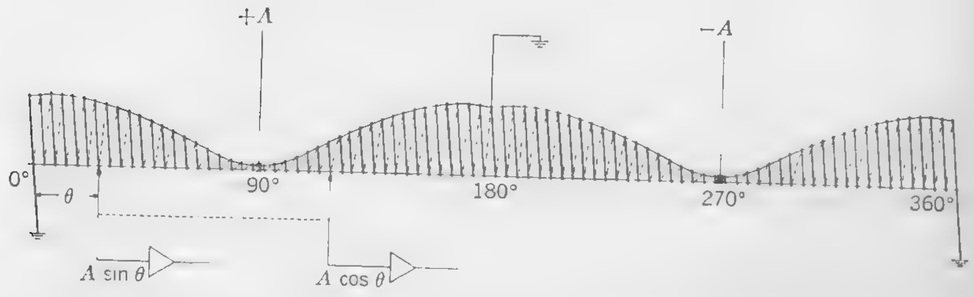
In RECTANGULAR the resolver servo computes the coordinates of a vector subjected to a rotation of axes (Fig. 3.1.2). In order to use the resolver for such a problem it is essential to know the gear ratio between the follow-up potentiometer and the resolver shaft. This ratio is the *resolver scale factor* K_r, usually expressed in degrees per volt. The permissible excursion of the resolver shaft angle θ should be at least 360 deg.



(a)



(b)



(c)

FIG. 3.1.20. Wire-wound d-c sine-cosine potentiometers. (*a*) Rectangular potentiometer. The rectangular resistance winding is rotated by the input shaft. The axis of rotation is perpendicular to the plane of the diagram. The four contacts are attached to the fixed potentiometer case; each contact travels over the card in a circle about the axis of rotation. Two rectangular cards are required for a resolver. (*b*) Rotary potentiometer-variable pitch. The sine function is approximated by straight-line segments, seven sections per quadrant for 0.125 per cent conformity. The resistance wire is wound on a mandrel with

since the normal follow-up travel is ±100 volts, the resolver scale factor must be no less than 1.8 deg/volt. A convenient and often used value for K_r is 2 deg/volt. The resolver is usually aligned with respect to the follow-up so that with zero servo input the resolver is at zero degrees and the resolver angle takes on positive and negative values. Then with $K_r = 2$ the range of θ is essentially ±200 deg; there is an overlap of 10 deg at 180 deg. Although the resolver itself need not have stops, the servo requires limit stops in this mode to restrict the travel to slightly less than the nominal values (see Art. 3.1.6).

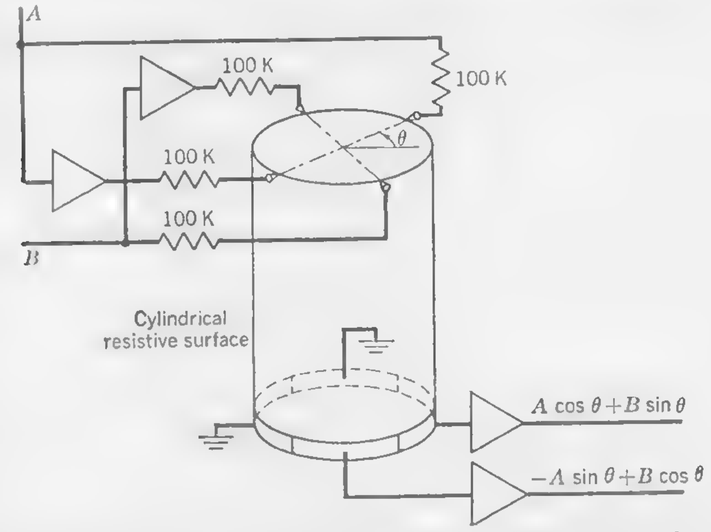


FIG. 3.1.21. Film resolver. The operation of this resolver depends upon the current distribution of a uniform, thin resistive cylindrical surface with boundary conditions determined by the four brushes at the top and the segmented conductive ring at the bottom (see Ref. 18).

In RECTANGULAR the resolver output accuracy depends not only upon the conformity error but also upon the accuracy with which the resolver shaft is positioned. Assuming that the mechanical errors between the follow-up and the resolver (e.g., backlash, shaft runout, mounting tolerances and shaft eccentricities) can be neglected, the resolver position error is $\Delta\theta = \epsilon K_r$, where ϵ is the error in volts of the follow-up position. The resolver output errors caused by this angular error ($\Delta\theta$) depend upon the angle θ and the resolver electrical inputs A and B. These errors are evaluated using the same method as before (see the definition of conformity): let one of the electrical inputs, say A, equal zero. Then the maximum errors in the α and β outputs are

$$\Delta\alpha = \Delta\beta = \epsilon K_r B/57.3 \text{ volts} = 0.873\epsilon K_r \text{ per cent}$$

where the resolver error is taken as a percentage of full scale, $2B$ volts. For the α output the maximum error occurs at $\theta = 0$ deg, and for β it occurs at $\theta = 90$ deg.

---

circular cross section. The mandrel is divided into several segments and a different winding pitch (i.e., number of wire turns per inch) is used in each section. In order to provide adequate resolution the ratio of the maximum to the minimum pitch is usually less than 3 or 4 to 1. Since the sine function requires a larger slope change, shunt padding resistors are used to reduce the slope. In practice the number of shunt resistors depends upon the manufacturing methods and the winding tolerances. Two mandrels of the type shown are needed for a single-turn sine-cosine potentiometer and two concentric sine-cosine potentiometers (four mandrels) are required for a single-turn resolver. (*c*) Rotary potentiometer-shaped card. The length of resistance wire for each individual turn is equal to twice the width of the card at the location of that turn; the rate of change of output with shaft position (gradient) is proportional to the width of the card. The shape of the card is modified from the theoretical form to correct for a 1-megohm load. The card shown is wrapped into cylindrical shape; two concentric cards are required for each resolver.

Under steady-state conditions the servo static error is usually negligible, and the resolver position error is $\epsilon = 2LK_r$ degrees where L is the follow-up linearity in per cent and the voltage across this potentiometer is ± 100 volts. Nonlinearity of the follow-up is a practical limitation on the accuracy of the resolver output; this error may be as large as or even larger than the resolver conformity error. For example, if $L = 0.025$ per cent and $K_r = 2$ deg/volt, the static resolver error is 0.087 per cent. If the gear ratio K_r is increased to 20 (i.e., the angular range is increased to $\pm 2{,}000$ deg), the static error becomes 0.87 per cent. Thus increased resolver angular range is obtained at the expense of accuracy. These calculations are predicated on the assumption of zero or negligible loading of the follow-up; this can be achieved by a bridge-type input circuit such as Fig. 3.1.15*a*. Under dynamic conditions the position

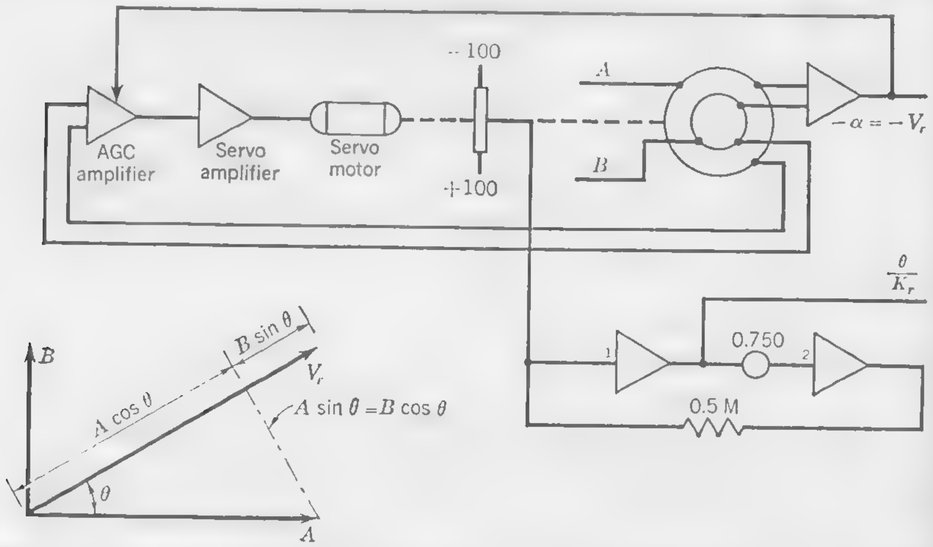


FIG. 3.1.22. POLAR operation. The inputs are A and B; the outputs are V_r and θ. The β output of the resolver is the servo error signal. An AGC amplifier is required to maintain servo performance independent of the inputs. The θ output is obtained from one of the servo linear potentiometers; an unloading circuit is used to prevent potentiometer loading errors. In lieu of this circuit it is possible to use a special potentiometer (at extra cost) which is linear for a prescribed load resistance.

error is larger because of the servo error (Fig. 3.1.15), and the resolver error increases in proportion.

POLAR operation of a resolver servo is used for the computation of a vector from its two components. If the components are A and B, the magnitude of the resultant is $V_r = \sqrt{A^2 + B^2}$ and its phase angle is $\theta = \arctan B/A$. These calculations are performed indirectly. The angle θ is determined by using the β output of the resolver as the servo error signal; the servo drives its error signal to a null. With $\beta = 0$, the α output will be equal to R. If the phase angle θ is also required, a signal proportional to this angle can be obtained from the output (arm) of one of the servo linear potentiometers with a constant voltage across its terminals. For accurate results, the potentiometer output should be unloaded (Fig. 3.1.22).

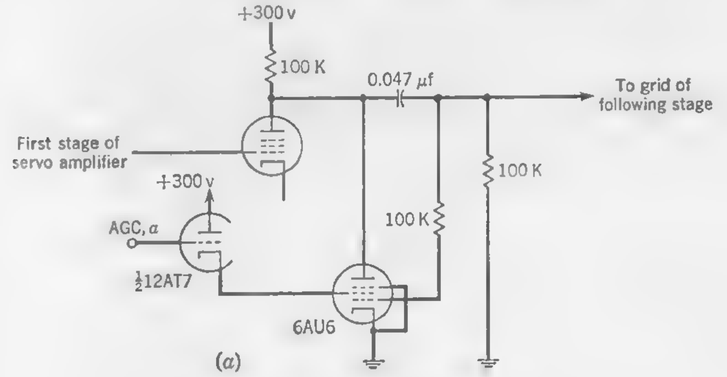
In POLAR the components in the servo loop are the same as for the servo multiplier except that the follow-up voltage gradient, $K_p = d\beta/d\theta$, is now dependent upon the problem variables A and B.

$$\frac{d\beta}{d\theta} = -A \cos \theta - B \sin \theta = -\alpha$$

This gradient may have a large amplitude range, varying typically from 100 volts/rad to 1 volt/rad. In order to obtain satisfactory servo performance; the loop gain must

be made independent of α. This is usually accomplished by providing an automatic-gain control (AGC) amplifier with a gain varying inversely with α. (Art. 3.1.13). This amplifier is connected in tandem with the servo amplifier.

If the resolver servo has single-turn linear potentiometers, the limit stops serve no useful purpose in POLAR. Servos with *retractable limit stops* permit the resolver to operate without limit on the number of revolutions (continuous rotation). Mechanical



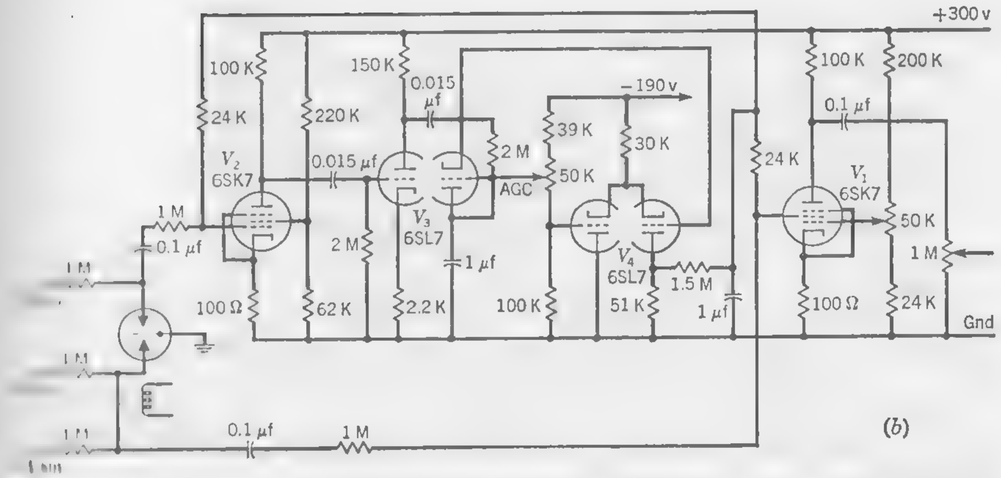


FIG. 3.1.23. AGC circuits using variable-μ pentodes. Method (*a*): The control pentode is used as a variable plate load. This circuit has limited amplitude range because the required functional relation (gain inversely proportional to α) depends upon the tube characteristics. Method (*b*): This circuit uses two identical variable-mu pentodes (V_1 and V_2). V_1 is used as an a-c amplifier whose gain is controlled by the feedback circuit consisting of V_2, V_3, and V_4. The AGC voltage, α, is modulated and used as input to V_2. The feedback circuit provides a d-c bias controlling the gain of V_2 in such a way that its output remains constant, independent of α. This same bias is then used for V_1. A potentiometer is provided in the screen circuit of V_1 to compensate for small differences in the characteristics of V_1 and V_2.

stops can be retracted by solenoid operation; electrical stops can be withdrawn by removing the limit voltages (see Art. 3.1.14). On the other hand, limit stops are always required for servos with multiturn potentiometers. It is sometimes advantageous to have a servo drive more than one resolver, particularly if it is a continuous-rotation servo. Then one resolver may be used in POLAR and the others in RECTANGULAR.

3.1.13. AGC Circuits. The type of AGC to be used depends upon whether the resolver servo has an a-c or a d-c loop. For a-c servos the servo error signal (the β output of the resolver) is modulated and combined with the d-c control signal (the α

output of the resolver). The latter is used to regulate a nonlinear element which varies the a-c gain of the amplifier. Finally, the a-c and d-c signals are separated. If the servo uses an induction resolver with the same carrier frequency as the servo, the a-c error signal can be obtained directly from the resolver rotor, bypassing the demodulator and modulator.

One commonly used nonlinear element is a variable-mu pentode; the transconductance of the pentode can be controlled by varying its bias. The α output of the resolver is used for the bias and thus varies the a-c gain. The simple circuit of Fig. 3.1.23*a* is effective for a limited range of α because the performance depends on the pentode characteristics. The amplifier of Fig. 3.1.23*b* uses two pentodes and a feedback circuit to improve the AGC performance. The servo loop gain is maintained constant within 6 db for a 26-db change in the control signal. The necessity of maintaining the circuit in adjustment is its principal disadvantage.

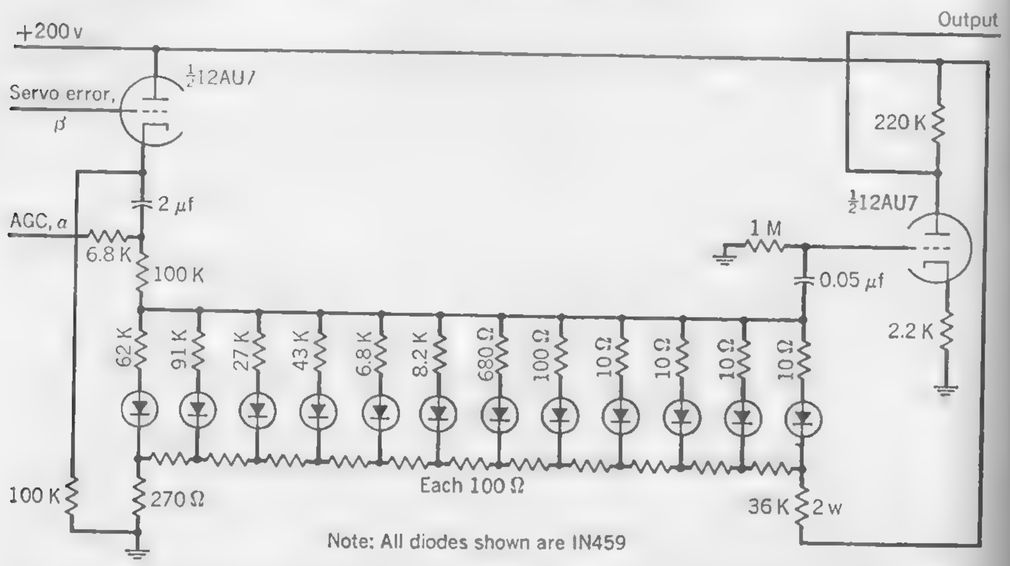


FIG. 3.1.24. AGC amplifier using diode function generator. The desired functional relation, gain inversely proportional to α, is obtained by a set of straight-line approximations. The servo error is modulated (not shown) and summed with α. This d-c voltage determines the number of diodes that conduct and thus the slope (or gain) of the function generator. The 0.05 μf condenser is used to block the d-c component in the output.

An AGC circuit that remedies these difficulties is shown in Fig. 3.1.24. The modulated error signal is summed with the d-c gain-control signal and fed to a diode-controlled attenuator. Its gain varies inversely with the control signal. The d-c component is removed at the attenuator output with a blocking condenser. For this circuit the servo loop gain is constant within 6 db for a 40-db change in the control voltage. The amplitude range of this circuit is restricted because the noise increases in proportion to the gain and eventually saturates the amplifier. In practice it is difficult to obtain an amplitude range in excess of 40 db.

These circuits cannot be used for d-c servos because both the error signal and the control signal are d-c. A different technique is required. One method is to use a high-gain d-c servo amplifier with nonlinear feedback. A feedback resistance which varies inversely with the control signal is required. A practical circuit based on this principle uses a photoconductive cell as the feedback resistance (Fig. 3.1.25). The resistance of the cell is inversely proportional to the light intensity to which it is exposed. The control signal α is connected to a neon lamp which is mounted close to the cell in a light-tight enclosure. The light intensity is proportional to this signal. The servo loop gain is maintained constant within 2 db for a 26-db change in the control signal.

3.1.14. Continuous Resolvers and Velocity Servos. As noted above, in the rectangular mode the resolver angle is usually restricted by the servo limit stops to essentially one revolution. This limitation is due to the type of servo used to drive the resolver and not to the resolver itself. The outputs of the resolver (α and β) are always bounded even if the angle θ increases without limit. A system capable of providing these outputs is called a continuous resolver. A signal proportional to the angular rate $d\theta/dt$ is available in many applications requiring a continuous resolver.

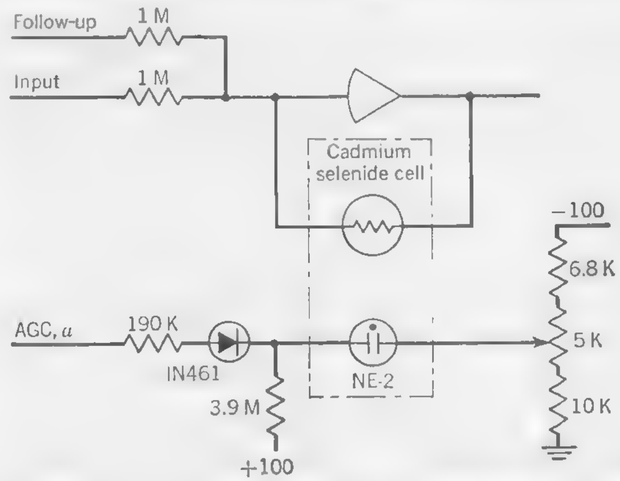


FIG. 3.1.25. AGC circuit for d-c servos. The d-c servo amplifier (see Fig. 3.1.9 and text) has a photoconductive cell as the feedback resistance. The magnitude of the AGC input, α, controls the light intensity of the neon and this light intensity is inversely proportional to the resistance of the cell. The +100-volt source is used to keep the neon fired at all times (Ref. 15).

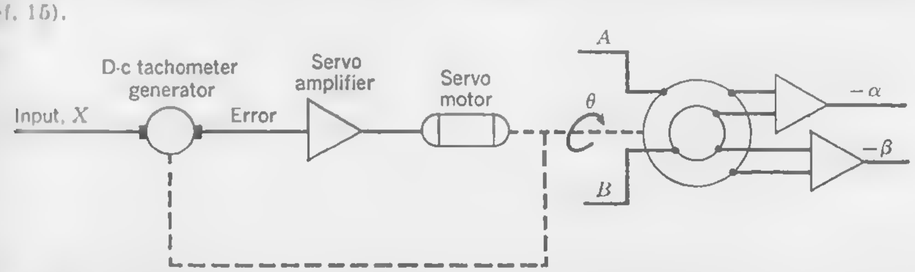


FIG. 3.1.26. Velocity servo. The tachometer is used as the feedback element; the servo error signal is the difference between the tachometer voltage and the input signal X. Under steady-state conditions, the servo error is zero and the input impedance is infinite. The shaft velocity is then proportional to X and the shaft position is proportional to the integral of the input signal.

Electromechanical and electronic methods for producing a continuous resolver have been devised; only the former will be considered in this chapter. One electromechanical system uses a two-integrator sine-wave circuit and one or two multiplying servos in the circuit of Fig. 3.25*a* or *b*. In a second method, a velocity servo driving a resolver is used; a block diagram is shown in Fig. 3.1.26. The components in this servo loop and in the position servo (Fig. 3.1.1*a*) are the same, except that the feedback potentiometer is replaced by a d-c tachometer generator. The input signal X, therefore, controls the angular rate; in particular $d\theta/dt = cX$ where c is a constant.

The angular position is

$$\theta(t) = c\int_0^t X\,dt + \theta_0$$

where θ_0 is the initial position. For this reason a velocity servo is also called an *integrating servo*.

Since this servo is an integrator, it requires the same operating modes as the electronic computing integrators (Sec 4.1). To introduce the initial angle (in RESET) the servo is converted to a position mode with a linear follow-up potentiometer. A voltage proportional to the initial angle is fed to the initial condition (IC) terminal (Fig. 3.1.27). Since the angular position is to have unlimited rotation in OPERATE, only single-turn linear potentiometers may be used. Limit stops are required in RESET and then must be removed when the computer is switched to OPERATE.

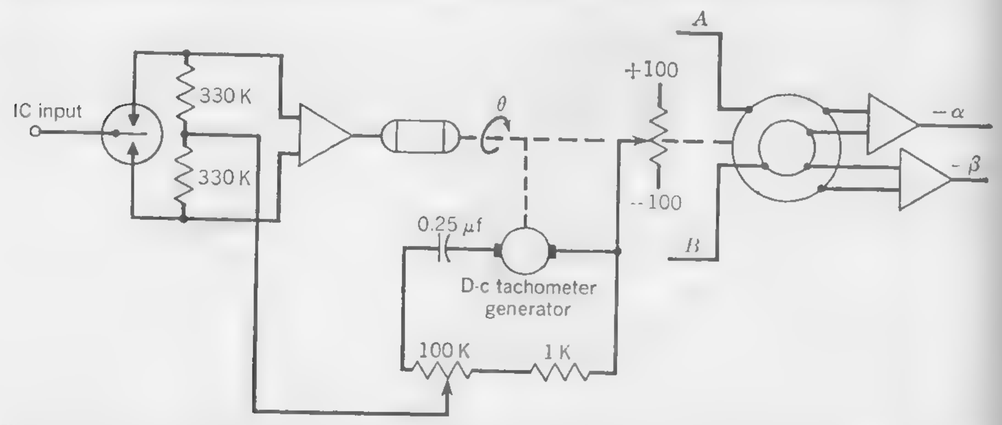


FIG. 3.1.27. Velocity servo in RESET. The servo is operated in its position mode; the shaft position θ is proportional to the IC input. In this mode the tachometer is used for damping.

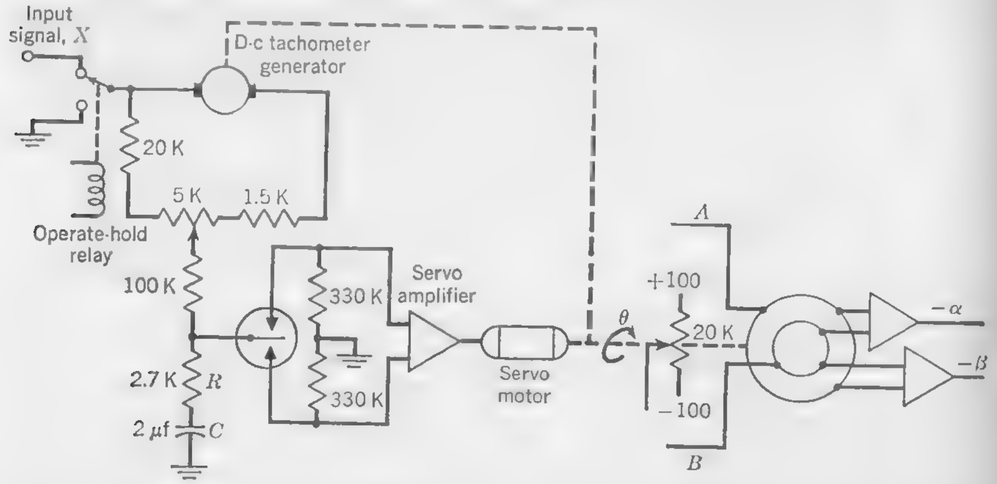


FIG. 3.1.28. Velocity servo in OPERATE and HOLD. In OPERATE the relay is deenergized and the shaft velocity is proportional to the input signal (see Fig. 3.1.26). The 5-kilohm potentiometer is used to adjust for differences in tachometer voltage gradient (see Table 3.1.6). In HOLD the relay is energized and the servo input is grounded. The RC network acts as a short-time-constant integrator and prevents a position error when the servo is switched to HOLD. The system inertia causes the servo to overshoot; the transient error is stored in the condenser and then the servo is driven back to the correct angular position.

In RESET, the follow-up potentiometer and the resolver zeros are aligned to permit θ_0 to take on both positive and negative values. The follow-up and the resolver must be geared one-to-one; otherwise their zeros will not remain in register after one revolution of the follow-up. The initial angle must be slightly less than ± 180 deg because of the gap in the follow-up, say ± 5 deg. The usual IC scale factor is 1.75 deg/volt. In addition to RESET and OPERATE it is usually desirable to provide a HOLD mode for the velocity servo. This is easily accomplished by removing the input signal and grounding the servo input (Fig. 3.1.28). Since this servo is an integrator

and it has inertia, there will be errors in starting or stopping the servo. To minimize these errors an RC integrating network is connected to the servo summing junction.

The performance of a velocity servo depends to a large extent upon the characteristics (especially the linearity) of the tachometer. Typical electrical specifications are given in Table 3.1.6 and a sample linearity curve is shown in Fig. 3.1.29. The characteristics of the other elements of the servo system—the servo amplifier, linear potentiometer, servo motor, and mechanical components—have been described in the preceding section. The important differences in system design are the gain of the servo amplifier and the gear ratios. The limitations on the maximum amplifier gain, K_A, do not apply to a velocity servo in OPERATE (Art. 3.1.8). In this mode the servo is an integrator and there is no need to limit the gain of the amplifier. In practice K_A will be restricted by the input noise level; for a typical high-performance servo the amplifier gain is 100 db. In HOLD the servo is in a position mode and the amplifier gain must be reduced.

There are three basic gear ratios: the resolver to the follow-up potentiometer, the motor to the tachometer, and the motor to the resolver. As previously explained, the first of these gear ratios must be one to one. The motor to tachometer gear ratio, K_τ, is determined by matching the maximum speeds of the motor and tachometer. For example, if the motor of Fig. 3.10c,d is used with the tachometer of Fig. 3.1.29 this gear ratio is 1.6, so that when the motor is rotating at maximum speed (4,800 rpm) the tachometer is driven at its maximum velocity (3,000 rpm). Such close gearing of the tachometer to the motor reduces the servo dynamic performance, but in this instance the linearity of the integrator is of prime importance. Finally, the gear ratio from the motor to the resolver K_ρ is a compromise value determined by the same factors as the multiplying servo gear ratio, K_g (Art. 3.1.8). These three gear ratios determine the integrator scale factor or gain. Since the servo has two possible outputs, the resolver and the linear potentiometer, there are two corresponding definitions for the integrator gain. If the tachometer voltage gradient is V_τ in volts per rpm, then the integrator gain to the resolver is $60K_\tau/K_\rho V$ deg/sec/volt and to the linear potentiometer it is $60K_\tau/1.75K_\rho V_\tau$ volts/sec/volt. It is convenient for the operator if one of these scale factors is made a suitable integer (say 1, 5, or 10). This can usually be achieved by adjusting K_ρ.

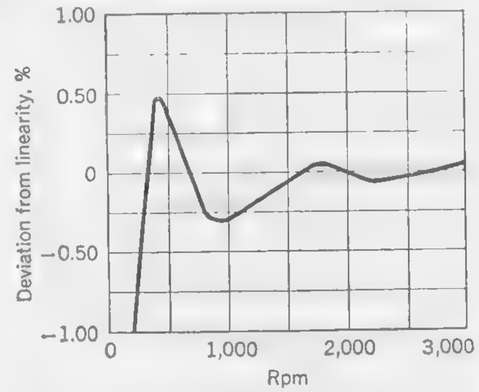


FIG. 3.1.29. D-C tachometer linearity characteristic. The linearity error is expressed as a percentage of the local value. For speeds between 100 and 200 rpm the linearity is approximately 2 per cent. For speeds below 100 rpm the deviation from linearity is less than 0.01 volt. At zero speed the output is zero.

By adding relays for control and linear potentiometers for multiplying, a computer velocity servo can be made extremely versatile. If the controls are wired so that each servo integrator may be individually controlled, i.e., the relays not permanently

Table 3.1.6. Electrical Specifications of a Typical D-C Tachometer

1. At 1,000 rpm the output shall be 24 ± 2.4 volts.
2. The output voltage shall be linear with respect to speed within ± 1 per cent of local value or 0.1 volt whichever is larger (see Fig. 31).
3. The maximum tachometer speed shall be 3,000 rpm.
4. The rms value of the ripple in the output voltage shall not exceed 2 per cent of the d-c output.
5. The minimum brush life with the tachometer running at 3,000 rpm shall be 600 hours.
6. The d-c resistance between diametrically opposite bars shall be 286 ± 43 ohms.
7. High potential test: 500 volts at 60 cps from each terminal to the frame for one minute.
8. The insulation resistance to ground shall be at least 50 megohms between the terminals and the frame measured with 500 volts d-c.

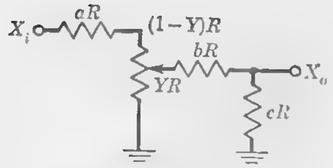
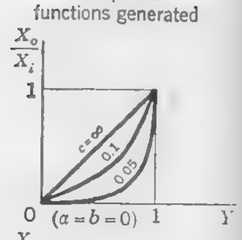
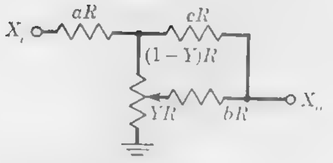
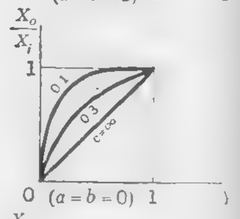
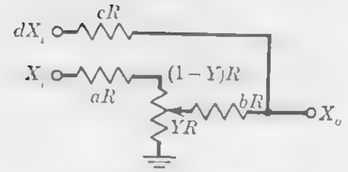
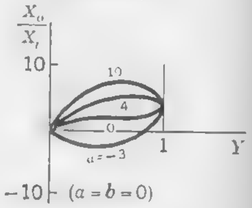
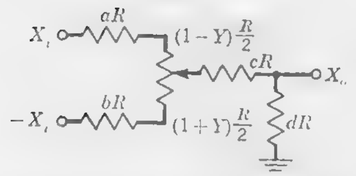
Circuit	Transfer function $\frac{X_o}{X_i}$	Examples of functions generated
	$\frac{cY}{(b+c)(a+1)+(a+1)Y-Y^2}$	
	$\frac{Y^2-Y(1+c)-b}{Y-Y(1+a)-a-(a+1)(b+c)}$	
	$\frac{Y^2-(a+1+\frac{c}{d})Y-b(a+1)}{Y^2-(a+1)Y-(a+1)(b+c)}\ d$	
	$\frac{4(b+Y-a)d}{[2(a+b)+4(c+d+ab+bd+ad+ac+bc)+1]+2(a-b)Y-Y^2}$	May be used for approximating tan Y ($a = b = c = 0$; $d = 0.448$)

FIG. 3.1.30*a*. Use of loaded linear function potentiometers for approximating various nonlinear functions. The potentiometer setting is determined by the machine variable Y. (*From Korn, G. A., and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

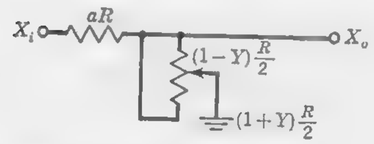
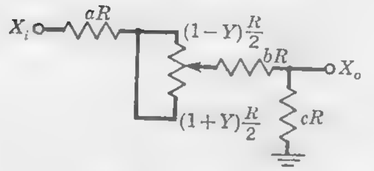
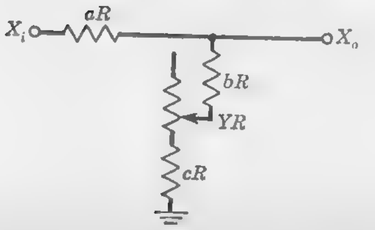
Circuit	Transfer function $\frac{X_o}{X_i}$	Examples of application
	$\frac{1-Y^2}{4a+1-Y^2}$	May be used for approximating $\frac{1-Y^2}{4a}$ (a large)
	$\frac{4c}{4(a+b+c)+1-Y^2}$	May be used for approximating sec Y ($a = b = 0$; $c = 0.25$)
	$\frac{b+c+Y}{a+b+c+Y}$	May be used for approximating $\frac{\sqrt{Y}}{1.79}$ ($a = 0.911$; $b = 0$; $c = 0.067$)

FIG. 3.1.30*b*. Miscellaneous circuit arrangements permitting the use of simple linear potentiometers for generating nonlinear functions. The networks are adjusted in each case to match the desired functions. The potentiometer setting is determined by the machine variable Y. (*From Korn and Korn, op. cit.*)

connected to the control lines (Sec. 4.1), the servo may be used as a position servo in connection with

1. A rectangular resolver
2. A polar resolver (an AGC amplifier is required) and
3. A multiplying servo

SERVO FUNCTION GENERATION

3.1.15. Survey. The analog circuit for a specified nonlinear function is not unique. A careful choice of the mathematical form and the circuit configuration is essential

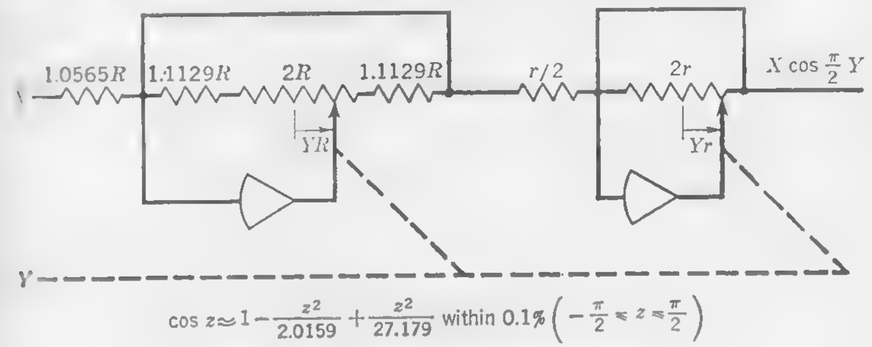


FIG. 3.1.31. Cosine computing circuit.[13]

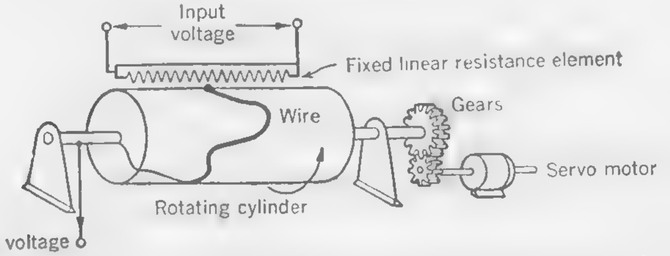


FIG. 3.1.32. Cam potentiometer. Linear resistance usually 20,000 ohms with 0.1 per cent linearity. Typical functional accuracy is 0.15 per cent of input voltage.

for accurate results. A simple example is a function with an appreciable linear content; then the linear portion can be removed and the nonlinear portion generated separately. Another technique that may be helpful in function generation is a transformation of coordinates.[17]

The most versatile servo function generators utilize *tapped potentiometers*. Other techniques for using servos as nonlinear function generators include:

1. *Polynomials.* The linear potentiometers of the servo can be used directly to generate polynomials; the highest degree polynomial that can be generated depends upon the number of potentiometers driven by the servo. If only odd powers of polynomials are needed, the use of center-tapped potentiometers will permit the saving of amplifiers.

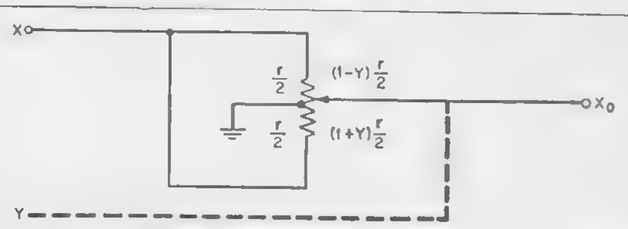
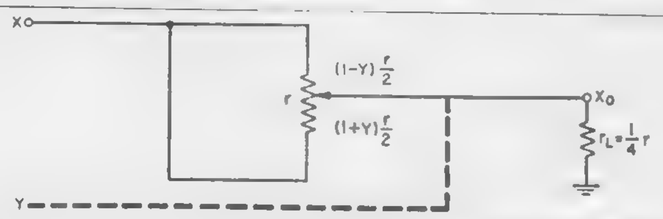
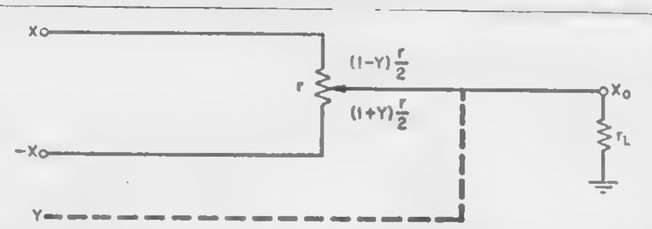
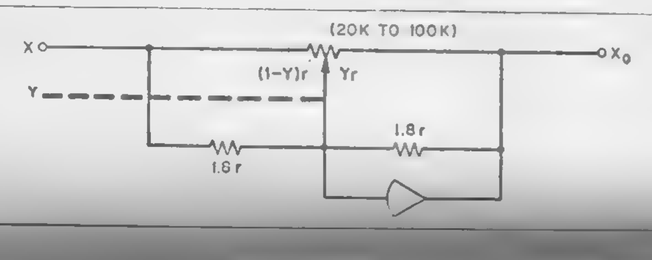
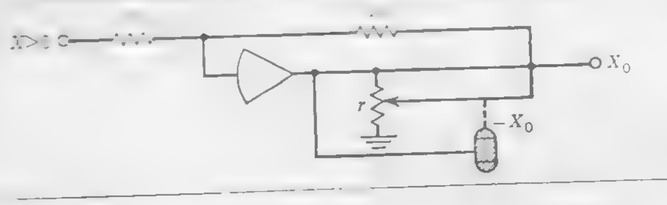
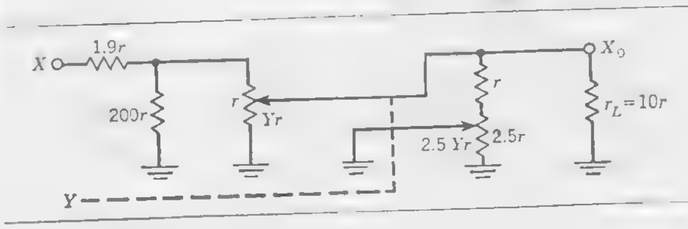
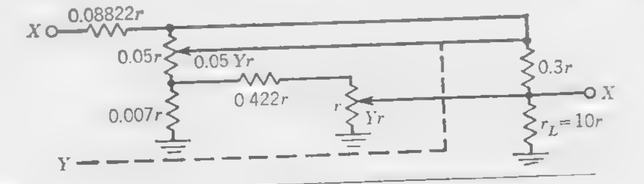
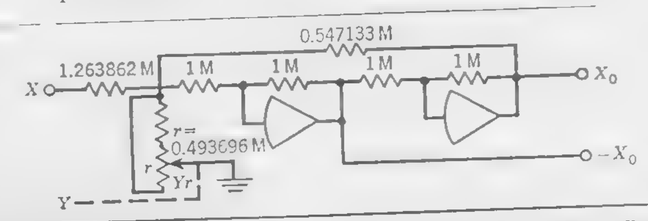
2. *Loading.* Functions which can be expressed as the ratio of two polynomials can be simulated by loading the linear servo potentiometers (Fig. 3.1.30 and Table 3.1.7).

3. *Variable Feedback.* Useful and accurate simulation of various functions can be obtained by using the servo linear potentiometers as input or feedback resistance of an operational amplifier (Fig. 3.1.31 and Table 3.1.7).[13,16]

4. *Special Potentiometers.* Nonlinear wire-wound or film potentiometers may be constructed for special-purpose applications. These are driven directly by the servo shaft.

5. *Cam Potentiometers.* The servo may be used to drive an insulated drum on which is mounted a paper carrying a wire cemented to it. A linear resistance element

Table 3.1.7. Generation of Functions with Linear Potentiometers (from Korn and Korn, *Electronic Analog Computers*)
r_L may be a load resistance. *Errors are computed for perfect resistors and potentiometers, negligible source impedances, and infinite-gain amplifiers.*

No.	Machine equation	Circuit or block diagram	Remarks
1	$X_o = X\lvert Y\rvert$		Subject to loading error, which may be corrected by loading the servo follow-up potentiometer
2	$X_o = X\frac{1}{2 - Y^2} \approx \frac{X}{2}\sec Y$		Error $\approx \frac{1}{24}Y^4$
3a	$X_o = X\frac{2Y}{3 - Y^2} \approx \frac{2}{3}X\tan Y$ for $r_L = 0.5r$		Angular error within $\pm 0.04°$ for $\lvert Y\rvert \leq \frac{1}{2}$*
3b	$X_o = X\frac{1.792Y}{2.792 - Y^2} \approx \frac{1.792}{2.792}X\tan Y$ for $r_L = 0.448r$		Angular error within $\pm 0.1°$ for $\lvert Y\rvert \leq 1$*
4	$X_o = -X\frac{(2.8 - Y)Y}{(1.8 + Y)(1 - Y)} \approx -X\tan\frac{\pi}{2}Y$		Angular error within 0.034° for $0 \leq Y < 1$†
5	$X_o = -\sqrt{\frac{R_2}{R_1}}X \qquad X > 0$		$r \ll R_o$
6a	$X_o = 0.2X\sin\frac{\pi}{2}Y$		Error less than 1% for $0 \leq Y \leq 1$‡
6b			Error less than 0.4% for $0 \leq Y \leq 1$‡
7	$X_o = 0.242718\,X\sin\frac{\pi}{2}Y$		Error less than 0.004 for $0 \leq Y \leq 1$‡

* Hofstadter, R., A Simple Potentiometer Circuit for Production of the Tangent Function, *Rev. Sci. Instr.*, **17**: 298, 1946.
† Seay, P. A., An Accurate Tangent Computing Circuit, *REAC Newsletter*, August, 1954 (Reeves Instrument Corporation, New York City).
‡ *Project TYPHOON Summary Progress Report No. 4*, Vol. 2, RCA Research Laboratories, Princeton, N.J. (unclassified).

is mounted parallel to the drum axis and tangent to the drum. The wire on the curve acts as the arm of the potentiometer and picks off a portion of the voltage applied across the potentiometers. The output depends upon the shape of the wire curve (Fig. 3.1.32).

6. *Automatic Curve Followers.*

TAPPED LINEAR POTENTIOMETERS

3.1.16. Construction. *Wire-wound servo potentiometers* have taps welded to one turn of wire on the resistance element. The tap locations and the number of tap connections are limited only by the physical space occupied by the tap-terminal lugs. Tap resistance (typically 5 ohms) is negligible compared to the potentiometer resistance per section.

Servo film potentiometers are available with two types of taps:

1. *Finite width and zero tap resistance.* This method is not suitable for general-purpose computers, because at each tap there will be a flat spot in the output function.

2. *Zero width and finite tap resistance.* The resistance between the wiper and the tap when the wiper is at the tap location is approximately 4 per cent of the potentiometer resistance, i.e., 40 per cent of the resistance per section for a unit with 10 sections.

The usual number of taps is such that 10 to 20 linear segments are provided; typical accuracy of tap location is 0.5 deg. Loading (padding) resistors are connected to the taps and used to produce the desired function. These padding resistors may be small variable potentiometers or fixed resistors and may be mounted on interchangeable plug-in turrets to permit the operator to change functions rapidly.

3.1.17. Loading Circuits. The padding resistors control the voltage at each tap so that the output voltage is a piecewise linear approximation to an arbitrary function. Within any segment the potentiometer interpolates linearly between the tap voltages. The accuracy with which a given function can be approximated depends upon its slope and curvature as well as the number of tapped segments and the potentiometer load. The problem of properly padding the potentiometer can be divided into two parts:

1. Given the function and the tapped potentiometer determine the desired tap voltages, i.e., the best straight lines.

2. Given the desired tap voltages, determine the circuit configuration to produce these outputs accurately. Only this second problem will be treated here.

Two basic circuits for padding a tapped potentiometer are: tap-to-tap loading and tap-to-end loading (Fig. 3.1.33). In the first a loading resistor is connected across each segment to reduce the voltage gradient across that segment to the value required by the slope of the function. In the second method a loading resistor is connected from each tap to one end of the potentiometer. In each of these circuits the ± 100 volts may be replaced by an independent variable $\pm Z$ and the output will be $z = Zf(x)$, a special function of two variables. The first method has the advantages of minimum current drain from the source (this is particularly significant if the ± 100 volts are replaced by the outputs $\pm Z$ of d-c amplifiers) and little danger of burning out the servo potentiometer. Its disadvantage is that it can only be used for monotonic functions. The second method requires that each tap be fused; it can be used to generate monotonic or nonmonotonic functions.

A combination of the two methods is a reasonable compromise, using tap-to-tap loading on the monotonic sections and tap-to-end loading where the slope changes sign (Fig. 3.1.34).

3.1.18. Calibration. There are three techniques for determining the padding resistors for the desired output. The first is a low-cost practical method in which the resistance values of the padders are computed; then the resistors are adjusted to the calculated values using, for example, a Wheatstone bridge. The calculations are a simple and direct application of Kirchhoff's laws at each tap position. The function is divided into monotonic sections. The calculations always start with the segment that has the steepest slope; this segment is unpadded and its current is determined immediately from the given slope. The calculations then proceed in turn to succeeding segments.

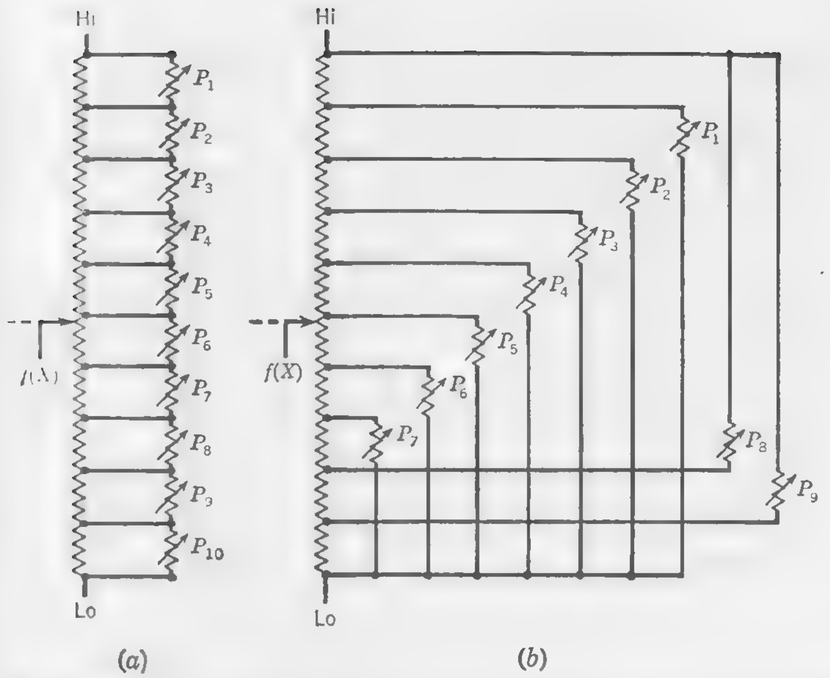


Fig. 3.1.33. Tapped linear potentiometer for function generation. (*a*) Tap-to-tap loading. (*b*) Tap-to-end loading.

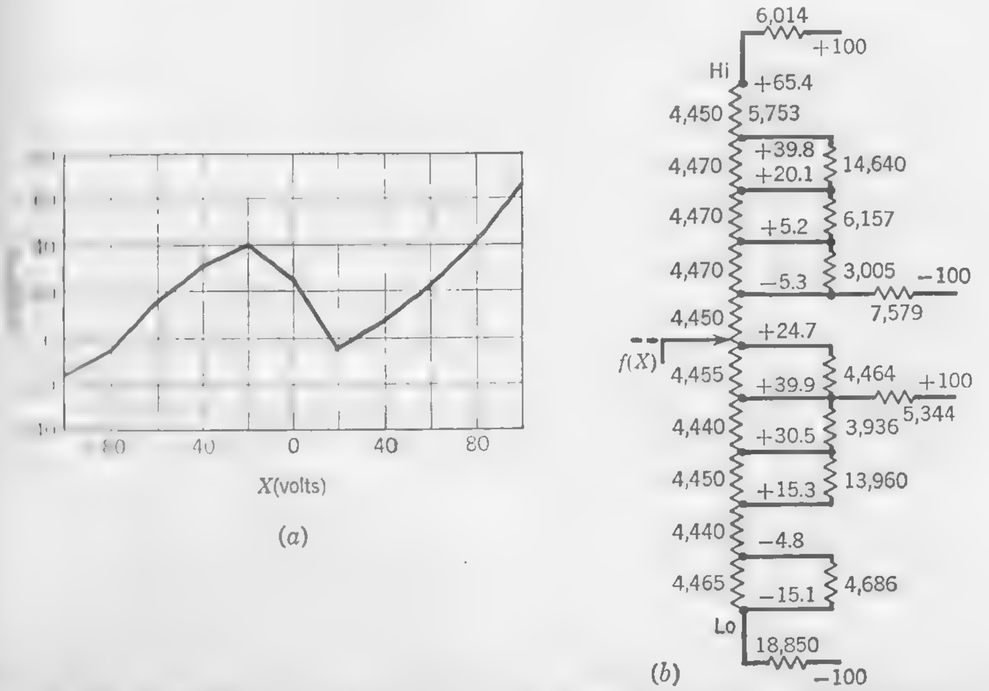


Fig. 3.1.34. Generation of a desired function by means of linear tapped potentiometer using both tap-to-tap and tap-to-end loading. (*a*) Desired function. (*b*) Potentiometer with padding resistors. Desired voltage at each tap location is shown.

At the taps where the slope changes sign tap-to-end loading of the appropriate polarity is required (Fig. 3.1.34).

In the second method, the servo is driven in turn to consecutive tap locations and at each tap the neighboring padding resistor is adjusted until the output becomes equal to the desired value. The inherent shortcoming of this method is that in general there will be interaction between the padding resistors. If, for example, the servo is returned to tap 1 after potentiometers $P1$ and $P2$ (Fig. 3.1.33) have been adjusted the output at this tap will no longer be correct. This means that a long and tedious iteration procedure is required and thus this method is usually not practical. The one exception to this rule is for monotonic functions driven by a constant current source (Fig. 3.1.35).

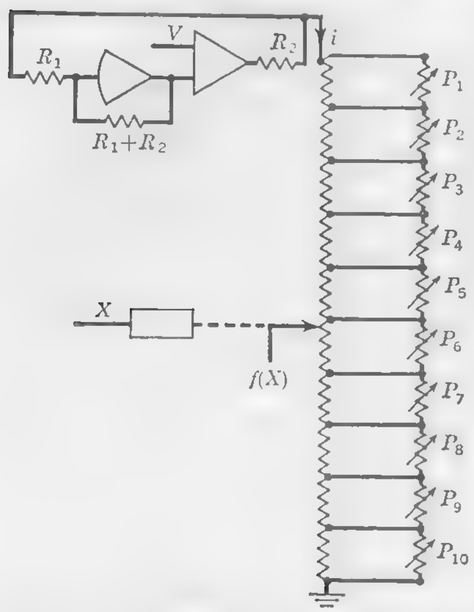


FIG. 3.1.35. Tapped linear potentiometer driven by current source. A positive-feedback circuit is used to generate a current, $i = V/R_1$, through the tapped potentiometer. If r is the resistance of the padded potentiometer one must have

$$r < \frac{R_1 R_2}{R_1 + R_2}\left[\frac{100}{|V|} - 1\right]$$

to prevent the amplifier output from exceeding 100 volts.

In the third method the padding resistors are also adjusted empirically but the interaction is prevented by using a special calibration device. As before the voltage at consecutive tap locations is measured and at each tap the neighboring padding resistor is adjusted until the output equals the desired value. As each tap is calibrated, the padder is removed from the next tap and replaced by the desired voltage applied by means of a d-c amplifier (Fig. 3.1.36).

3.1.19. Practical Considerations. 1. *Tap Location and Follow-up Loading.* Since equal spacing of the taps is only nominal, the actual servo inputs at the tap positions must be determined by direct measurement for a specified loading. This data should be recorded in a suitable log book for each tapped potentiometer for standard loading resistances applied to the follow-up.

2. *Tapped Potentiometer Loading.* Since padding resistors are adjusted with the load connected, the only effect of the load resistance is to make the interpolation between taps slightly nonlinear (Ref. 12).

3. *Resistance between Taps.* If the first calibration scheme is used, the actual resistance between taps must be measured; these values should also be recorded in the log book.

4. *Tap Resistance.* All the calibration schemes described above assume negligible tap resistance. If the tap resistance is appreciable, an iteration procedure is required to determine the padding resistances.

5. *Resistance Value.* The resistance of the tapped potentiometer should be as large as possible, preferably at least 50K. After calibration the effective potentiometer resistance will usually be 20 to 25 per cent of the original value.

6. *Maximum Slope.* The power rating of the potentiometer limits the maximum slope m. Let the servo input voltage for full travel of the tapped potentiometer be $2E$ in volts, its power rating P in watts and its resistance r in ohms; then the maximum permissible slope is $m = \sqrt{Pr/2E}$ volts per volt. *Caution*: If an attempt is made to simulate a slope larger than this value, the potentiometer may burn out. For larger slopes the function must be rescaled.

7. *Typical Accuracy.* With the potentiometer properly loaded the over-all accuracy can be made as good as 0.05 per cent of the reference voltage (100 volts).

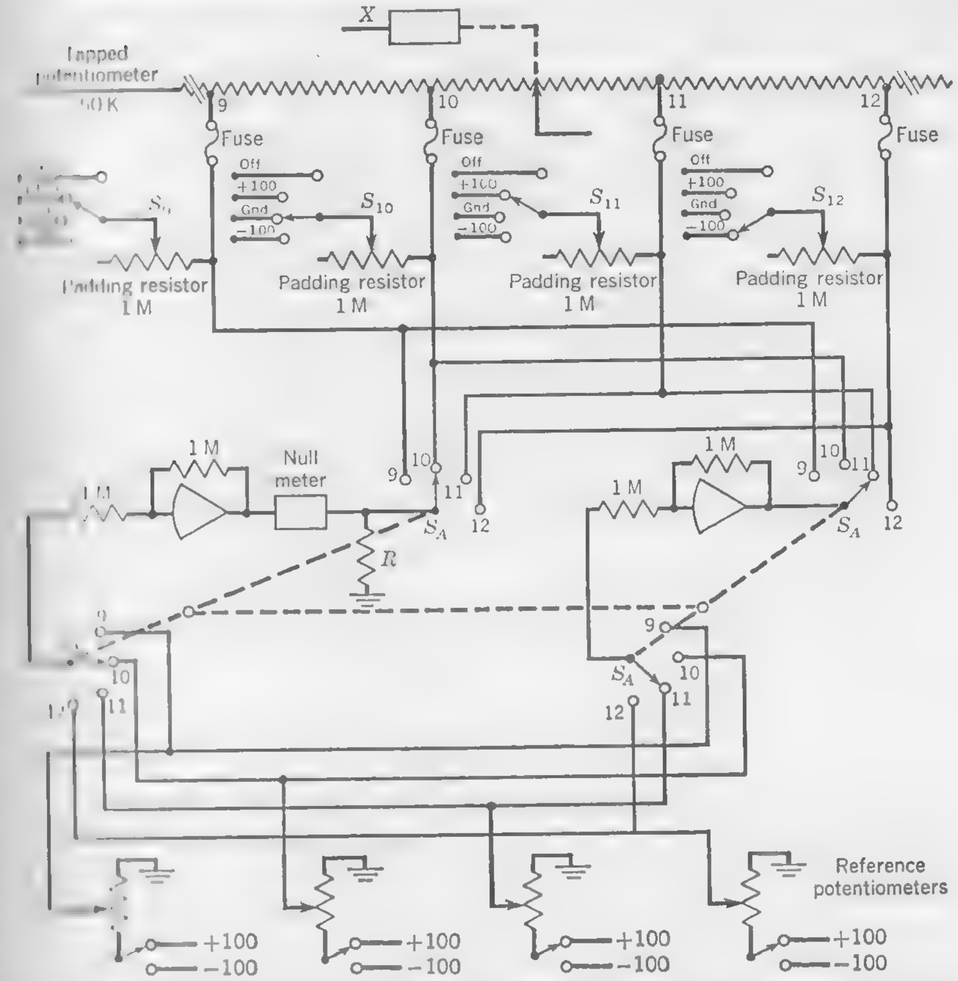


FIG. 3.1.36. Calibration circuit for padding tapped linear potentiometers. To calibrate tap 10: (*a*) Reference potentiometers are set to desired tap voltages. (*b*) Taps one to nine are assumed to have been calibrated. (*c*) Switch S_A is in position shown. (*d*) Switch S_{11} is turned to OFF position. (*e*) The padding resistor at tap 10 is adjusted until null meter is zero. Note that load resistor R is connected to the tap that is being calibrated.

3.1.20. Functions of Two Variables. Because of the linear interpolating feature of tapped potentiometers, they may also be used for generating functions of two variables $f(X,Y)$. The taps are fed by single variable function generators with outputs $f(X_1,Y)$, $f(X_2,Y)$, $f(X_3,Y)$, where $X_1, X_2, X_3, \ldots$ are fixed values of X. As before the function $Z\,f(X,Y)$, a function of three independent variables, can easily be generated (Fig. 3.3.19).

References

1. Blernson, G. A.: Comparison of Lead Network, Tachometer, and Damper Stabilization for Electric Servos, *IRE Trans. on Automatic Control*, vol. AC-4, no. 1, May, 1959.
2. Bradley, F. R., and R. D. McCoy: Computing with Servo-driven Potentiometer, *Tele-Tech*, September, 1952.
3. Bromberg, E.: Functional Potentiometer, Patent No. 2,554,811, May, 1951.
4. Brown, G. S., and D. P. Campbell: *Principles of Servomechanisms*, John Wiley & Sons, 1948.
5. Chestnut, H., and R. W. Mayer: *Servomechanisms and Regulating System Design*, vol. I, John Wiley & Sons, Inc., 1951.

6. Chestnut, H., and R. W. Mayer: *Servomechanisms and Regulating System Design*, vol. II, John Wiley & Sons, Inc., 1955.
7. Gilbert, E. O.: The Design of Position and Velocity Servos for Multiplying and Function Generation, *IRE Trans. on Electronic Computers*, vol. EC-8, no. 3, September, 1959.
8. Godet, Sidney: The Gated Amplifier Computer Technique, *Proceedings of National Simulation Conference*, Dallas, Texas, pp. 30.1–30.6, January, 1956.
9. Greenwood, I. A., J. V. Holdam, and D. MacRae: *Electronic Instruments*, MIT Radiation Laboratory Series, vol. 21, McGraw-Hill Book Company, Inc., New York, 1948.
10. Hogan, I. J.: Electrical Noise in Wire Wound Potentiometers, *West Coast IRE Convention*, August, 1952.
11. James, H. M., N. B. Nichols, and R. S. Phillips: *Theory of Servomechanisms*, McGraw-Hill Book Company, Inc., New York, 1947.
12. Korn, G. A.: Design and Construction of Universal Function Generating Potentiometers, *Review of Scientific Instruments*, **21**:77, 1950.
13. Lazarus, H.: Cosine Computing Circuit, Patent No. 2,852,191, September, 1958.
14. Mayer, Arthur: Electromechanical Resolver, Patent No. 2,864,924, December, 1958.
15. Morrill, C. D., and R. C. Weyrick: An AGC System for DC Servos, *Control Engineering*, p. 111, July, 1958.
16. Seay, P. A.: An Accurate Tangent Computing Circuit, *REAC Newsletter*, August, 1954.
17. Simulation Council Newsletter, *Instruments and Control Systems*, p. 451, March, 1960.
18. Staff Report: Electrical Resolvers, *Electromechanical Design*, pp. 49–71, February, 1960.
19. Van Voorhis, S. N.: *Microwave Receivers*, p. 402, McGraw-Hill Book Company, Inc., New York, 1947.

PART 2: ELECTRONIC MULTIPLIERS AND RELATED TOPICS

By CHARLES D. MORRILL

INTRODUCTION

3.2.1. Survey. Electronic analog multiplication methods may directly instrument a *physical law* (Arts. 3.2.2 and 3.2.3), use physical circuit elements or special function generators in *logarithmic* or *quarter-square multipliers* (Arts. 3.2.4 to 3.2.6), or provide multiplication through a combination of *modulation schemes* (Arts. 3.2.7 to 3.2.10). Some of these techniques may be improved through *multiplication using shared or similar variable transmission networks* (Arts. 3.2.11 to 3.2.13). Multipliers using the quarter-square and feedback-stabilized modulation schemes have been most successful in general-purpose analog computers. These multipliers provide four-quadrant operation; accuracies to within 0.1 to 0.01 per cent of full scale and bandwidths up to 2 to 20 kc are common. Generally, the higher accuracies are obtained at the expense of bandwidth reduction or increased noise. Output noise in multipliers that do not employ repetitive waveforms may be on the order of 0.01 volt rms. Multipliers such as the time-division multiplier may have output noise up to 0.1 volt rms. The effect of such increased noise is in some instances negligible, even in following multipliers, because of noise frequency and waveform; time-division multipliers have been used to raise a variable to the thirty-second power without any noticeable error attributable to the output noise of each stage.

Electronic dividers (Art. 3.2.14) and most *electronic resolvers* (Art. 3.2.15) are basically related to multipliers and multiplication schemes.

MULTIPLIERS BASED ON INSTRUMENTATION OF PHYSICAL LAWS

3.2.2. Multipliers Employing Electrical and Magnetic Fields. In the so-called *dynamometer multipliers*[1,2] a feedback loop balances forces or torques due to several electromagnets so that an output current is made proportional to the product of two input currents with an accuracy of 0.2 to 1 per cent. Dynamometer multipliers are electromechanical devices and do not compare with electronic multipliers in speed or accuracy.

In the *crossed-fields electron-beam multiplier*,[3] the electron beam in a cathode-ray tube is deflected electrostatically and electromagnetically by two signals. A counterbalancing electrostatic deflection requires a third signal proportional to the product of the first two signals (Fig. 3.2.1).

The *Hall-effect multiplier*[4] is a solid-state analog of the crossed-fields electron-beam multiplier. Accuracy to 0.1 per cent and response to 20 cps may be achieved (Sec. 7.1). A similar effect that may be instrumented to produce multiplication is the so-called *Gauss effect* or *magnetoresistor effect* on the resistivity of bismuth wire (Fig. 3.2.2; see also Sec. 7.1).[5]

3.2.3. Miscellaneous Multiplication Schemes. *Strain-gauge multipliers* are based on the linear relation between resistance and deformation of strain-gauge elements; a feedback loop causes a loud-speaker element to balance a strain-gauge bridge.[6] Accuracies to 0.2 per cent and frequency response to 200 cps have been obtained.

The use of *heat-transfer multipliers* (Sec. 8.3) and multipliers based on the photoresistive[7] and electrochemical[8] effects is restricted to special applications where either accuracy or speed is less important than simplicity. Similarly, multipliers that instrument the nonlinear characteristics of vacuum tubes (multigrid tubes or single-grid tubes with square-law characteristics), thyrite,[9] forward conduction of diodes,[10] special beam tubes,[11,12] and multipliers such as one deriving ratios from similar right triangles,[13] and products based on probability theory[14] have had limited use in special applications.

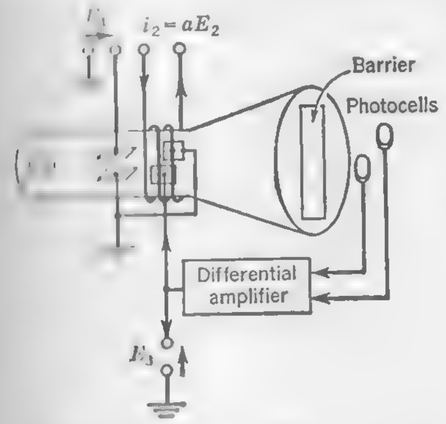


FIG. 3.2.1. Crossed-fields electron-beam multiplier.

$$E_3 = (K/\sqrt{E_a})E_1E_2$$

where E_a is the first anode voltage. Accuracy is within 1 to 2 per cent; response to changes in E_1 and E_2 are 5 to 10 cps, respectively (based on Ref. 3).

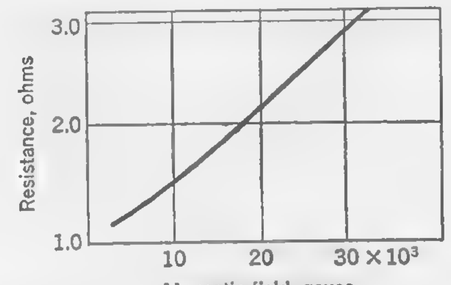


FIG. 3.2.2. Resistivity of bismuth wire in a magnetic field.

LOGARITHMIC, QUARTER-SQUARE, AND TRIANGLE-INTEGRATION MULTIPLIERS

3.2.4. Logarithmic Multipliers. Logarithmic multipliers implement the equation

$$\log^{-1}(\log X + \log Y) = XY \tag{3.2.1}$$

with the aid of various function generators (see also Sec. 3.3). Representative of "natural" logarithmic elements is the 9004 diode operating in the space-charge-limited region (Fig. 3.2.3).[15] When used in the feedback path of an operational amplifier, the logarithm of an input variable is produced, and vice versa. Logarithmic multipliers are inherently one-quadrant multipliers; four-quadrant operation is produced by adding positive constants to each input variable and subtracting extraneous terms from the results.

The logarithmic multiplier is inherently complicated and is not generally used. However, its elements may be useful in producing terms of the form $\log X$, $\exp X$, and $X^{1/m}$. In performing four-quadrant multiplication, an accuracy of about 1 per cent may be expected; bandwidth varies but is generally less than 50 cps.

3.2.5. Quarter-square Multipliers. *Quarter-square multipliers* implement a relation of the form

$$\tfrac{1}{4}\{[(X+Y)^2 + a(X+Y) + b] - [(X-Y)^2 + a(X-Y) + b] - 2aY\} = XY \tag{3.2.2}$$

While vacuum-tube characteristics, nonlinear resistors (thyrite, see Table 3.3.1),[9] and special beam tubes[11,12] have all been used to produce the required quadratic functions of the voltages $X + Y$ and $X - Y$, two types of quarter-square multipliers, based on *diode function generators* (see also Sec. 3.3) and *triangle integration*, are far superior to all others. The design of a commercially produced quarter-square multiplier employing biased-diode networks is illustrated in Fig. 3.2.4.

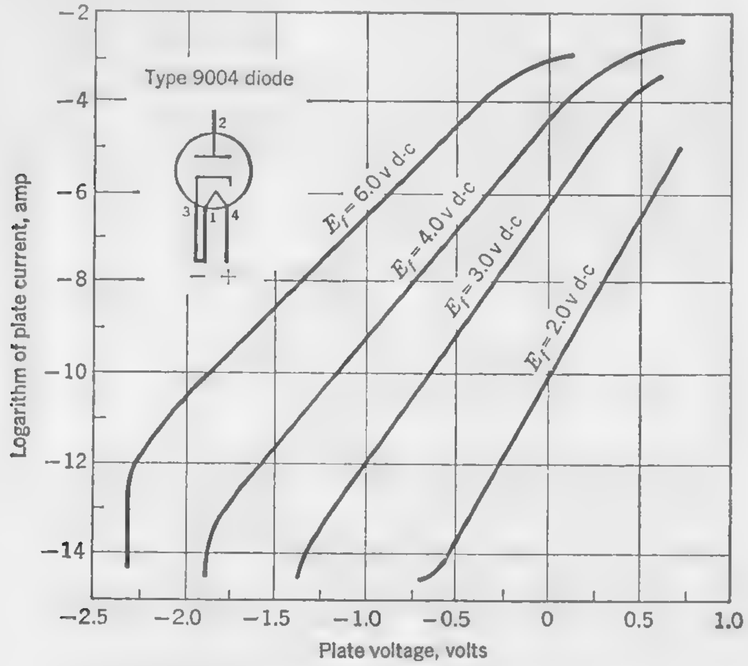


FIG. 3.2.3. Plate characteristics of a 9004 diode.

3.2.6. Triangle-integration Multipliers.[16,17,31 to 34] *Triangle-integration multipliers* generate the quadratic functions required to implement Eq. (2) by integration (averaging) of clipped triangular waveforms whose fundamental frequency can range between 1,200 cps and 2 Mc (Fig. 3.2.5). Since the triangular waveform must be accurately generated, clipped, and then filtered out, improvements in multiplier frequency response will involve compromises between static accuracy and output noise, as in modulation-type multipliers. In general, the multiplier output will be inversely proportional to the triangle amplitude, which must thus be accurately regulated.

Triangle-integration multipliers permit static accuracies of 0.5 volt with a frequency response yielding 5 deg phase shift at 2 kc, or 0.05 volt static accuracy with a proportionally smaller frequency range. With careful control of the frequency of the triangular waveform and use of a double-notched filter, noise may be reduced to between 0.01 and 0.05 volt rms. For low-accuracy multiplication or squaring, requirements on the linearity of the triangular waveform may be relaxed; thus replacement of the triangular waveform by a sinusoid can still yield a squaring circuit accurate within about 15 per cent. A single triangle generator (pulse generator plus integrator) can serve several multiplier channels.

Figure 3.2.6 shows an accurate triangle-integration multiplier based on the amplitude-selector circuit of Fig. 3.2.5. Simultaneous multiplication and division (see also

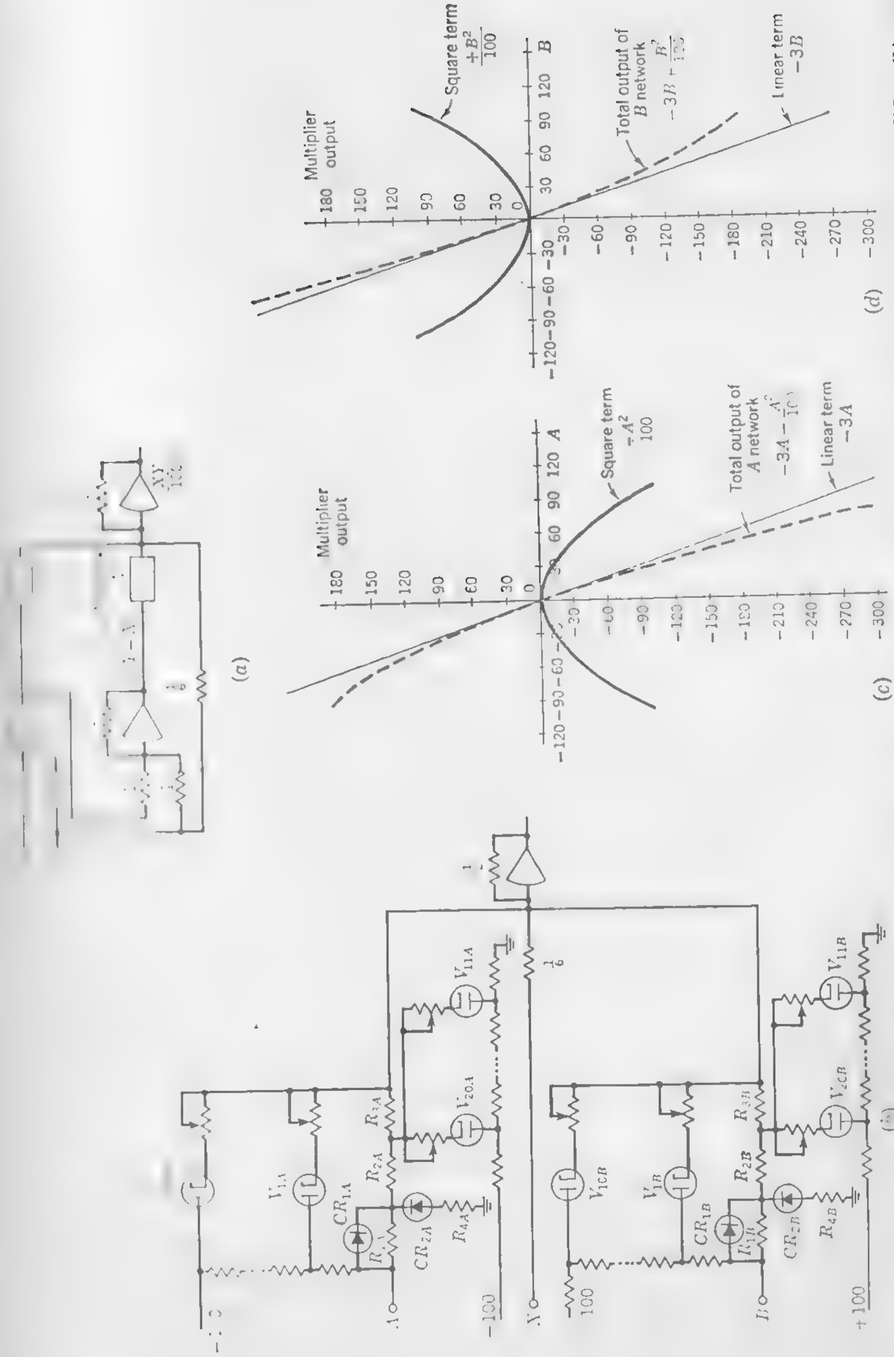


FIG. 3.2.4. Quarter-square multiplier employing biased diodes (*a*), details of networks *A* and *B* and output amplifier (*b*), and transfer characteristics of the *A* and *B* networks (*c*), (*d*). Virtually any operational amplifier may be used, an advantage of the method. Maximum error of about 0.25 volt and response beyond 1,000 cps are achieved. At 100 cps, phase shift averages about 1 deg and noise about 0.01 volt rms. (*Reeves Instrument Corp.*)

Art. 3.2.14) make the multiplier output independent of the triangle amplitude at the expense of some additional components.

The amplifiers labeled *W* are standard two-tube operational d-c amplifiers; those labeled *P* are chopper amplifiers used to provide stabilization against drift. This combination, along with current-booster circuits, is shown symbolically in amplifiers 1, 2, and 3, and in more detail in the circuits of amplifier 5. The booster of amplifier 6 provides additional voltage gain as well as current gain since this amplifier-booster assembly is operated at a net gain substantially greater than unity. The "index voltage decade" provides a convenient means of calibration and fault isolation. The essential feature of this multiplier is the diode-selection matrix shown in the upper part of the diagram. Each of the six diode pairs (*X*1-*X*2, *X*3-*X*4, etc.) selects the most positive of the two values connected to the diode anodes. Since points *D* and *E*

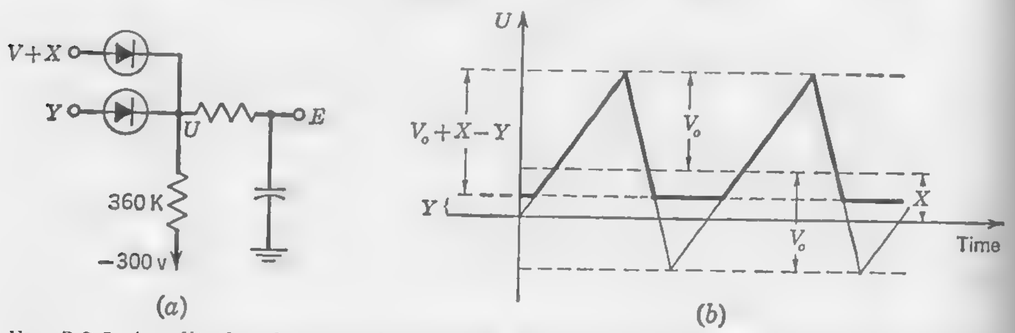


FIG. 3.2.5. Amplitude-selector and averaging circuits (*a*) and clipped triangular waveform (*b*). $V = V(t)$ is a triangular waveform of peak-to-peak amplitude $2V_0$. The average value of the amplitude-selector output U equals $(X - Y + V_0)^2/4V_0 + Y$.

are connected to summing points of operational amplifiers, they are at ground potential. Assuming diodes are perfect switches, reference to Fig. 3.2.5 shows that the average current through each summing resistor R is given by

$$i_{D_1} = \frac{1}{4V_0R}\left(\frac{e_1}{5} + V_0\right)^2 - \frac{e_1}{5R} \qquad i_{D_2} = \frac{1}{4V_0R}\left(\frac{e_3 - e_{out}}{5} + V_0\right)^2 - \frac{e_3}{5R}$$

$$i_{D_3} = \frac{1}{4V_0R}\left(-\frac{e_2}{5} + V_0\right)^2$$

$$i_{E_1} = \frac{1}{4V_0R}\left(\frac{e_1 - e_2}{5} + V_0\right)^2 - \frac{e_1}{5R} \qquad i_{E_2} = \frac{1}{4V_0R}\left(\frac{e_3}{5} + V_0\right)^2 - \frac{e_3}{5R}$$

$$i_{E_3} = \frac{1}{4V_0R}\left(-\frac{e_{out}}{5} + V_0\right)^2 \qquad (3.2.3)$$

The currents i_D are inverted by amplifier 4 and added to the currents i_E. The resultant currents into the summing junction of amplifier 5 must add up to 0, so that $e_{out} = e_1e_2/e_3$ with all voltages measured in volts. Static accuracy is within 0.1 volt, including long-term drift. A selector switch selects either a frequency response 3 db down at 10 kc with 10 deg phase shift at 1 kc or 0.5 deg phase shift at 500 cps with some sacrifice in amplitude response. Output noise is less than 0.035 volt rms (manufacturer's specifications).

Note that no amplifier in Fig. 3.2.6 is required to pass the high-frequency triangular waveform. If this requirement can be relaxed through the use of a lower triangle frequency (5 to 20 kc) and/or wide-band d-c amplifiers, then substantially simpler triangle-integration amplifiers can be assembled from the various limiter circuits described in Sec. 3.3 (Fig. 3.2.7; see also Sec. 7.1).[31,33,34] The precision limiter circuits of Art. 3.3.2 are of particular interest in this connection and permit multiplication accuracies of 0.05 volt with the circuit of Fig. 3.2.7*a*.[34]

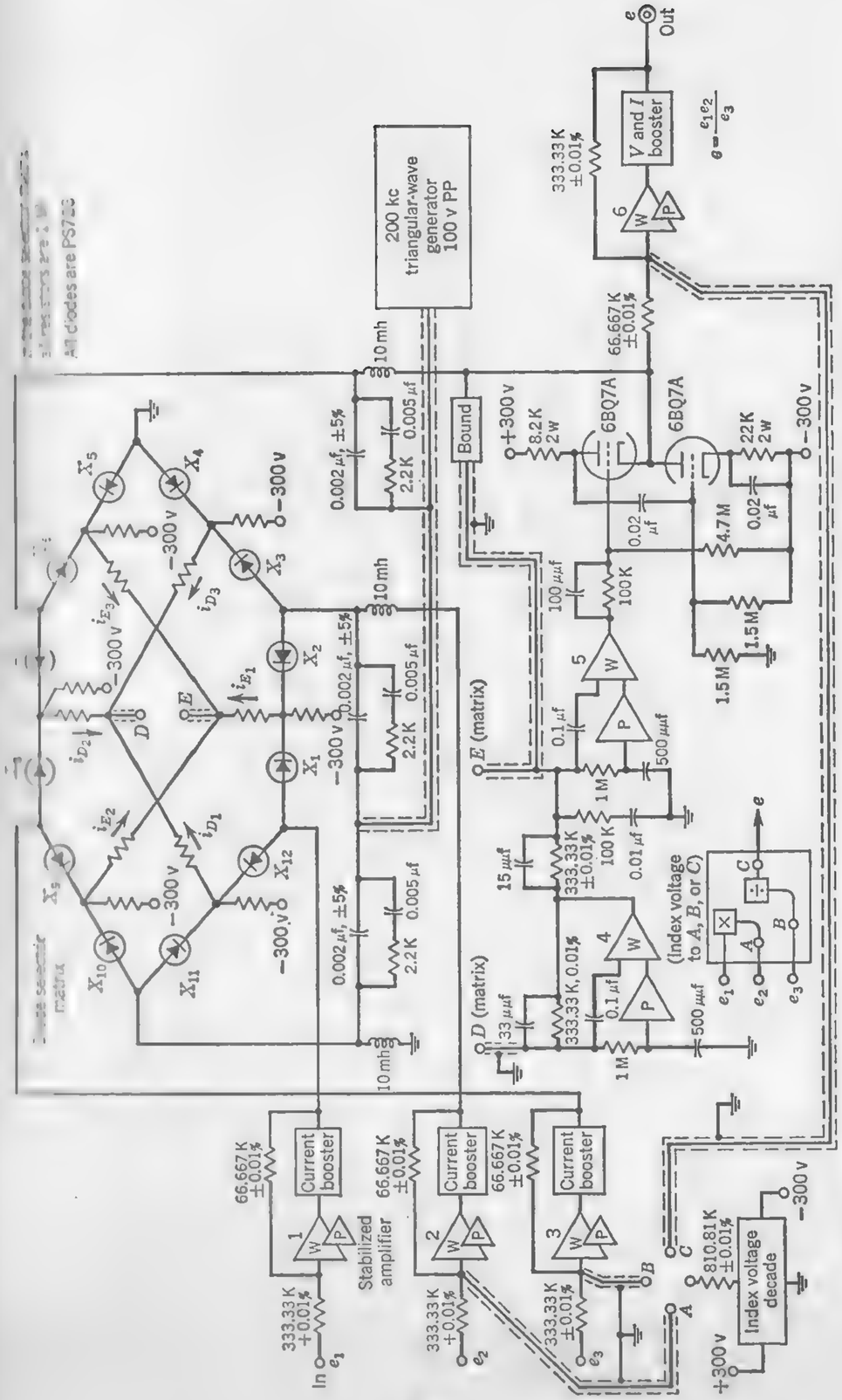


FIG. 3.2.6. An accurate triangle-integration multiplier/divider (Model K-5M, G. A. Philbrick Researches, Inc.).

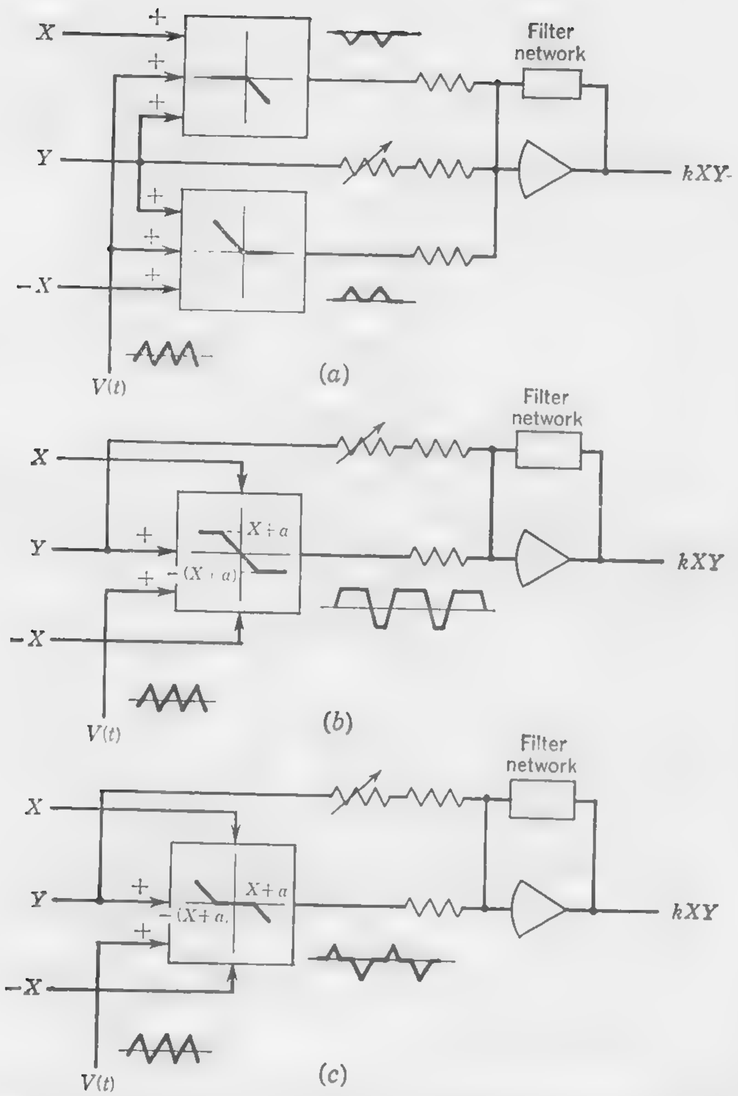


FIG. 3.2.7. Triangle-integration multipliers employing simple limiters (*a*), a dual limiter (*b*), and a dead-space circuit (*c*) (based on Ref. 34).

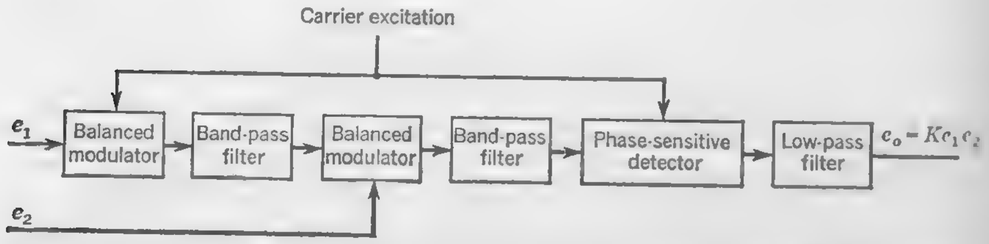
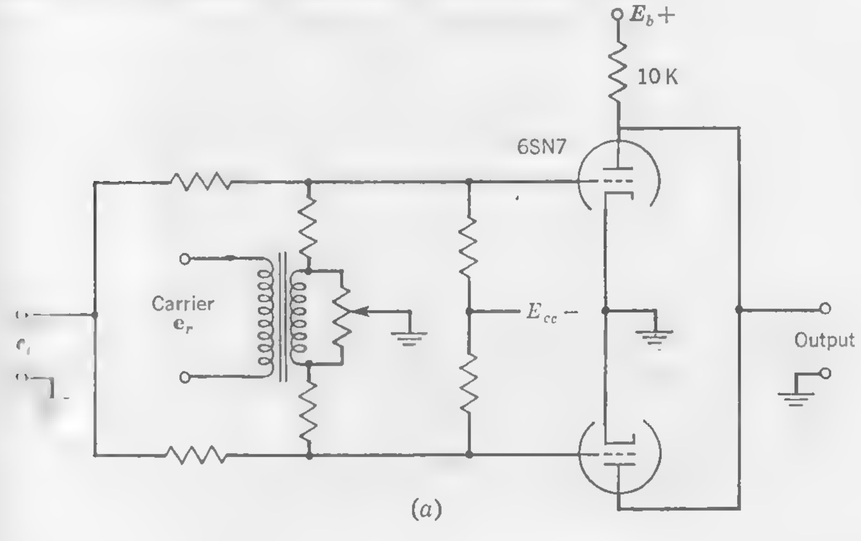


FIG. 3.2.8. Block diagram of an open-loop AM-AM multiplier. The second modulator must be balanced, since either input may be greater than the other. Depending on the rejection capability of the phase-sensitive detector, the bandpass filters may be omitted. An accuracy to about 3 per cent and a bandwidth of 70 kc have been reported.

MODULATION SCHEMES WITHOUT FEEDBACK REGULATION

3.2.7. AM-AM Multiplier.[18,19] In Fig. 3.2.8 a carrier is amplitude-modulated by one variable and serves as the carrier excitation to a balanced modulator (Fig. 3.2.9), where it is amplitude-modulated by a second variable. The synchronously detected (Fig. 3.2.10) and filtered result is proportional to the product. Although the circuit is basically simple, accuracy and stability are generally poor.



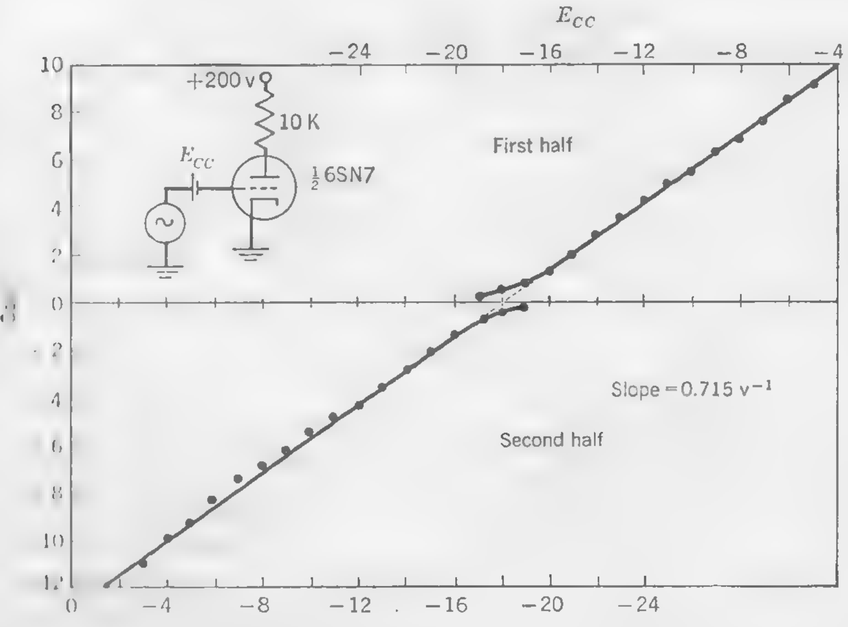


FIG. 3.2.9. A simple balanced modulator circuit (*a*), and its input-output gain due to each half triode (*b*). The output is seen to be approximately proportional to the square of the sum of e_i and $e_r = e_r \sin \omega t$. The portion of the output at ω is proportional to the product of e_i and e_r. Values of grid summing resistances are set so that, with $e_i = e_r = 0$, $(E_{cc})_1 = (E_{cc})_2 \approx -18$ volts, with e_i and e_r aiding, $(e_{ok})_1$ or $(e_{oc})_2 \approx -2$ volts. (*Goodyear Aircraft Corporation.*)

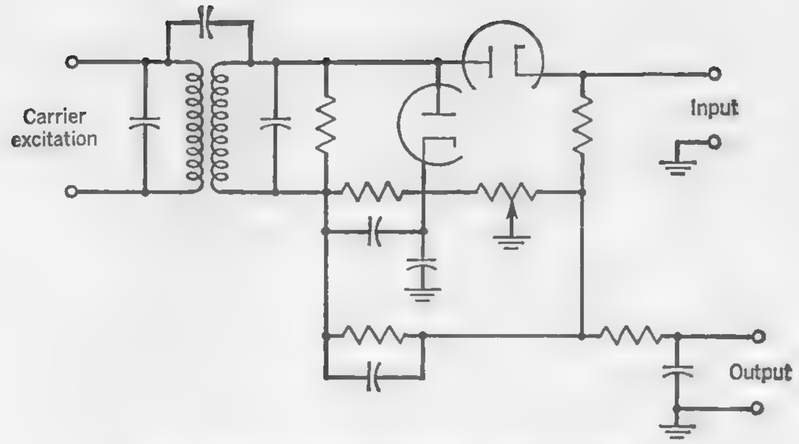


FIG. 3.2.10. A phase-sensitive detector circuit.

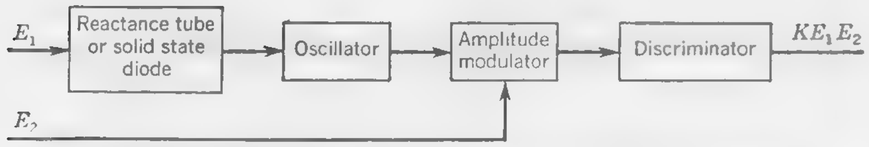
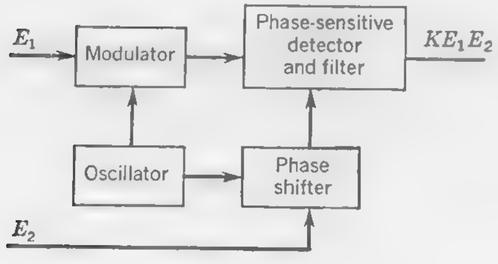
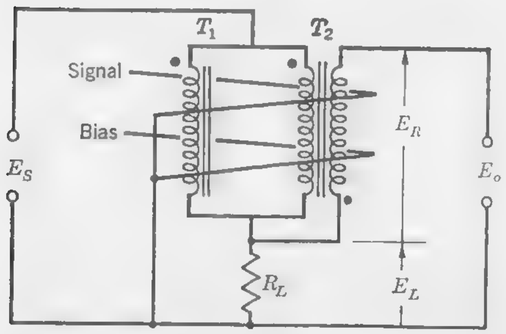


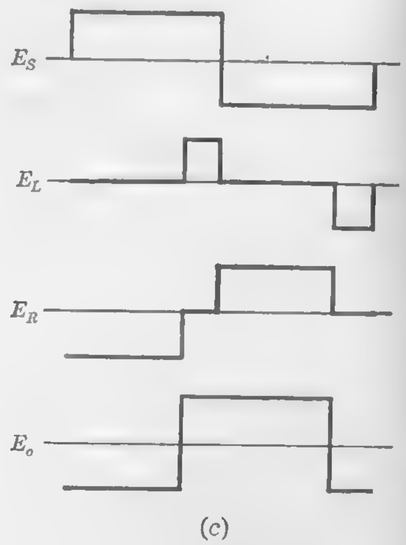
FIG. 3.2.11. Block diagram of an open-loop AM-FM multiplier.



(a)

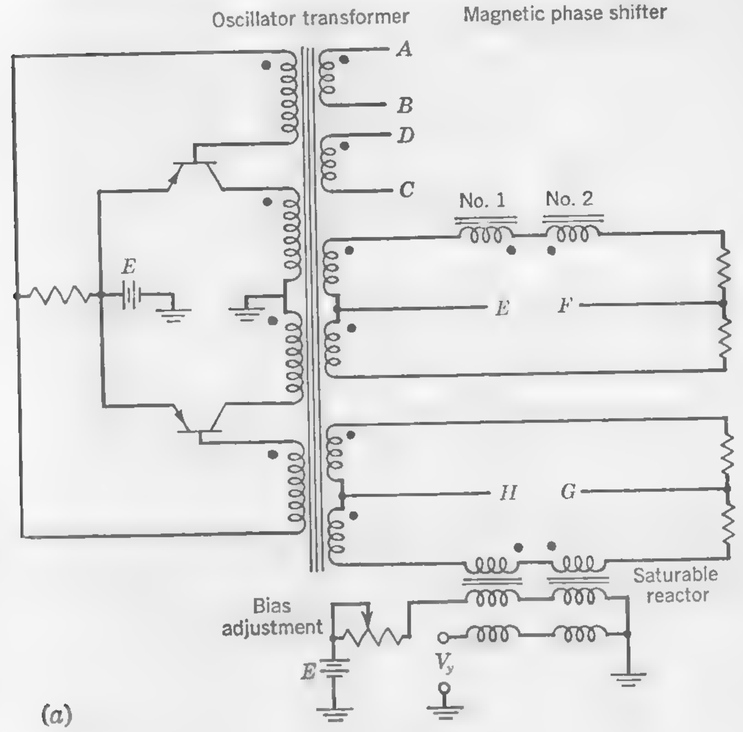


(b)

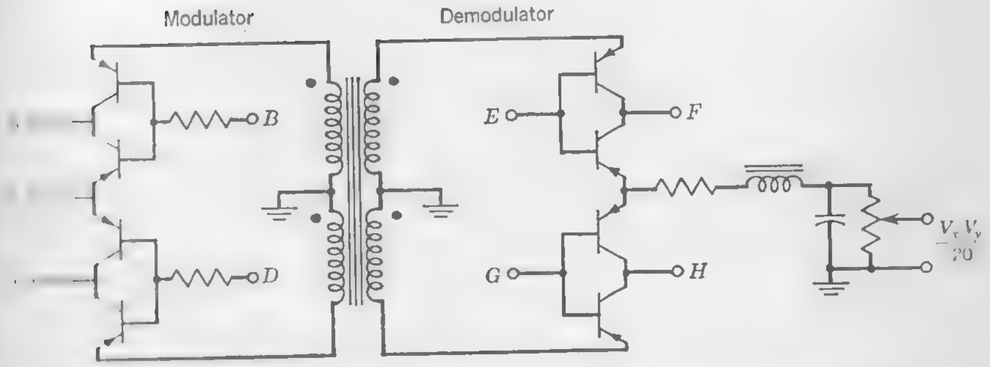


(c)

FIG. 3.2.12. Block diagram of an open-loop AM-PM multiplier (*a*), saturable-reactor phase shifter (*b*), and waveforms (*c*). In Fig. 3.2.12*b*, virtually the entire voltage appears across T_1 and T_2 (thus E_R) until either T_1 or T_2 saturates, at which time voltage appears across R_L. The saturation levels of T_1 and T_2 must be matched through the bias current adjustment, to the amplitude and frequency of E_S and value of R_L.



(a)



(b)

FIG. 3.2.13. Another magnetic-oscillator phase shifter (*a*), and modulator-demodulator (*b*). The principle of the resulting multiplier is the same as that shown in Fig. 3.2.12. Accuracy to about 3 per cent and bandwidth of about 100 cps may be obtained. In the referenced multiplier, inputs are ±10 volts, output is $(V_X V_Y)/20$, and the oscillator frequency 5 kc. Multiplier is partly self-compensating, since frequency as well as amplitude of phase-shifted detection signal varies with E.

3.2.8. Open-loop AM-FM Multiplier (Fig. 3.2.11).[20] The open-loop AM-FM multiplier has no advantage and some disadvantages, such as poorer linearity and stability, when compared with the AM-AM multiplier. Its one inherent advantage, a capability of separating the input effects on modulation products, is useful only in feedback schemes (Art. 3.2.11).

3.2.9. Open-loop AM-PM Multiplier (Figs. 3.2.12 and 3.2.13).[21,22] In terms of accuracy and response, the AM-PM multiplier seems to offer no advantage over AM-AM multiplication. However, its capabilities may not yet have been fully

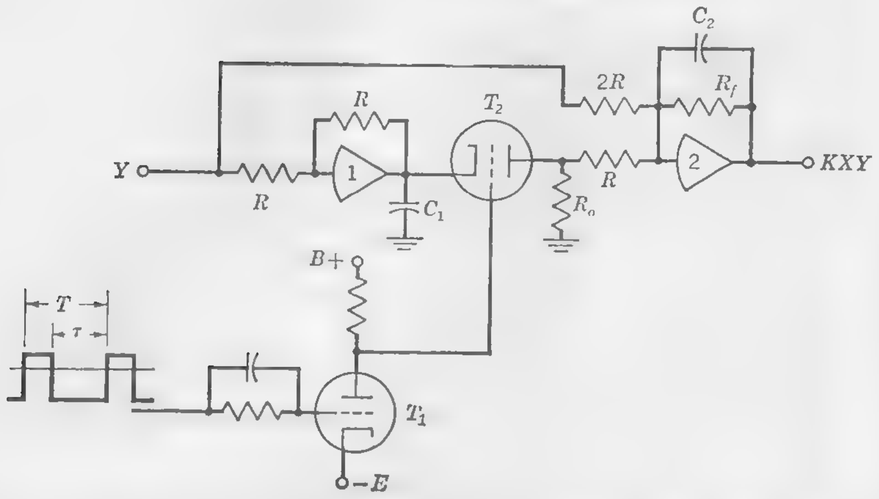


FIG. 3.2.14. Open-loop time-division multiplier. In operation, average output is proportional to both width and height of a rectangular waveform. Not shown is a pulse generator and a phantastron or equivalent circuit for producing time delays. Multiplier shown handles only positive quantities but is easily modified for four-quadrant operation with a resultant 4:1 increase in error voltage. T_2, driven from saturation to cutoff by T_1, acts as a switch. C_1 ensures low impedance at switching frequency. R_0 provides rapid discharge of stray capacitances. In one version, R_0 and the input through $2R$ are replaced by a second switched amplifier with input $-Y$ and output connected to the junction of R and the plate of T_2. R_f and C_2 provide filtering of the switching noise. The value of K, usually 0.01, depends on R_f, R, and the design center and adjustment of the phantastron. Accuracy to about 0.2 per cent and response to about 1 kc are typical.

exploited, and it seems well adapted to magnetic-amplifier techniques. The output is sensitive to changes in either amplitude or frequency of the excitation waveform.

3.2.10. Open-loop PAM-PWM Multiplier (Figs. 3.2.14 and 3.2.15).[23] The PAM-PWM or time-division multiplier (or pulsed-attenuator multiplier) has substantially better accuracy than most other modulation multipliers because of (1) the precision with which time delays may be produced[24] and (2) the accuracy with which rectangular waveforms may be amplitude-modulated. Sources and corrections of errors are discussed below.

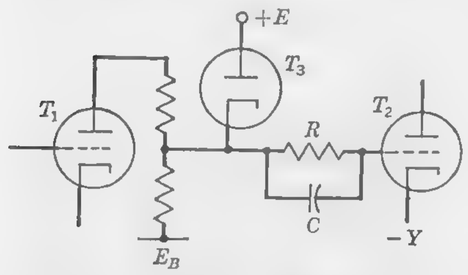


FIG. 3.2.15. Modified switch driver. A modification of the circuit of Fig. 3.2.14 to reduce the effect of finite switching speed. With C small, it charges rapidly during the first instant of ON time to $(E_B + Y)$; it discharges only a small amount through R when switch is off, but the amount is more than sufficient to track Y. The peak-to-peak amplitude of the keying waveform is now $(E_B - E)$ with peak value $-Y$, and changes in Y do not affect on-off ratios.

The switch tube T_2 (Fig. 3.2.14) when conducting may be approximated by a series battery and resistance. The effect of the equivalent battery may be eliminated by a small positive bias added to amplifier 1. One can eliminate both effects by placing the switch tube inside the forward loop, or by driving the switch with a current generator (Art. 3.2.12). In electronic switches the instants of opening and closing may be dependent on the voltage being switched, because of the finite rise rate of the switching waveform. Without compensation, error will be

$$\text{Per cent error} = \frac{E}{aT} \times 10^2 \qquad (3.2.4)$$

where E = voltage to be switched
a = average of rise and fall rates $(E = 0)$
T = period of switching waveform

In electronic multipliers employing voltage switches, this error may amount to 0.02 to 0.05 per cent. A method of minimizing this error is shown in Fig. 3.2.15.

Error is further introduced because of the coupling of the switching signal into the load R_0 (Fig. 3.2.14) through the grid-plate capacitance of T_2 when T_2 is switched off. Resultant voltage across R_0 is approximately

$$e_{av} = E_{sw}\frac{R_0 C_{gp}}{T} \qquad (3.2.5)$$

where E_{sw} is the voltage applied to the grid of T_2 having a grid-plate capacitance C_{gp}. If E_{sw} and T are constant, as in some multipliers, the error may be eliminated by simple addition. If either is nonconstant, capacitance coupling from an alternate switch driver or complementary-waveform source may be provided.

MULTIPLIERS EMPLOYING FEEDBACK REGULATION OF DUAL VARIABLE-GAIN DEVICES

3.2.11. Feedback Regulation of Gain.[32] The most accurate electronic multipliers use the feedback scheme of Fig. 3.2.16 to make the gain K of a voltage-sensitive variable-gain device proportional to the ratio X_2/Y of two voltages. K is independent of

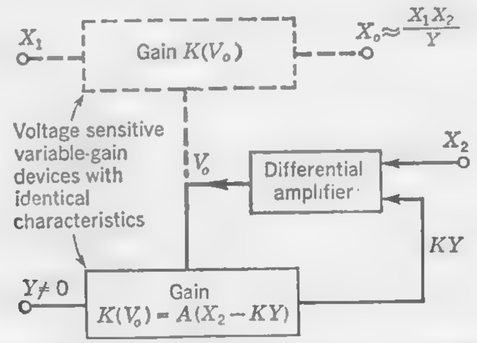
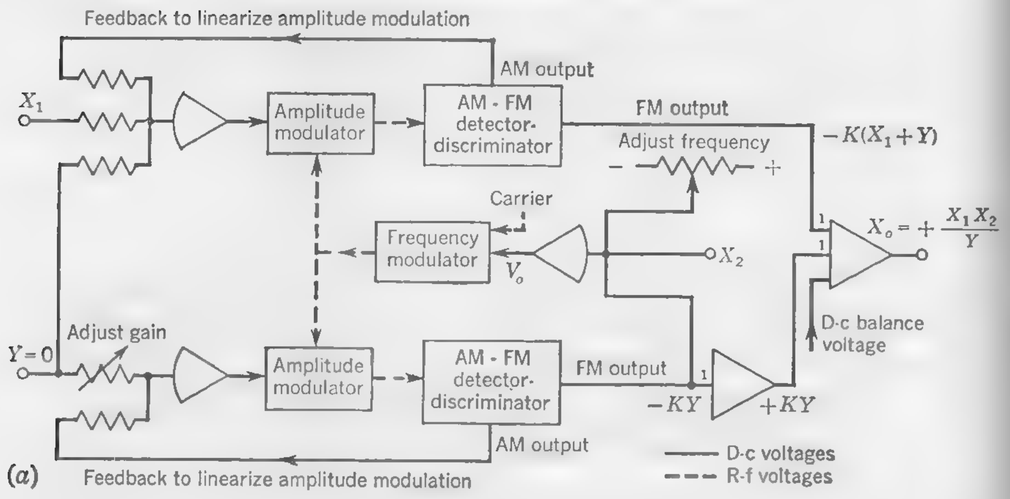


FIG. 3.2.16. Feedback scheme used for setting the gain of a variable-gain device accurately to a value determined by the input voltages X_2 and Y. Note that X_1, X_2, Y, and X_0 are measured in "machine units" of, say, 100 volts. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York*, 1956.)

X_1 or Y but varies monotonically (not necessarily linearly) with the gain-setting voltage V_0. Feedback through the differential amplifier nulls the difference $K(V_0)Y - X_2$, so that for high-feedback loop gain $K = X_2/Y$. *If the two variable-gain devices shown have accurately identical gains* $K(V_0)$, then the output is proportional to the desired product X_1X_2. Additional "slave channels" can yield other products X_3X_2, X_4X_2, ...

The best-known example of this feedback scheme is the well-known *servomultiplier* (Sec. 3.1); the *heat-transfer multiplier* described in Sec. 8.3 also operates on the same principle. Because of the difficulty in obtaining identical gain functions $K(V_0)$, several workers have proposed *frequency-sharing and time-sharing circuits employing a single variable-gain device.*[24,32] Most modern feedback multipliers employ two variable-gain devices based on modulation schemes, which do not depend critically on tube or semiconductor characteristics. Examples of such circuits are the *feedback-regulated AM-FM multiplier* (Fig. 3.2.17),[26] the *step multiplier* (Sec. 7.1), the *sampling multiplier* (Fig. 3.2.18),[25] and the *feedback-regulated pulsed-attenuator* or *time-division multipliers.*

3.2.12. Time-division Multipliers.[27,28,29,32] Feedback-regulated time-division multipliers can be *externally triggered* or *self-excited.* Constant triggering rates permit more selective filtering providing lower output noise and phase shift in the useful frequency band. Self-excited time-division multipliers are generally simpler.



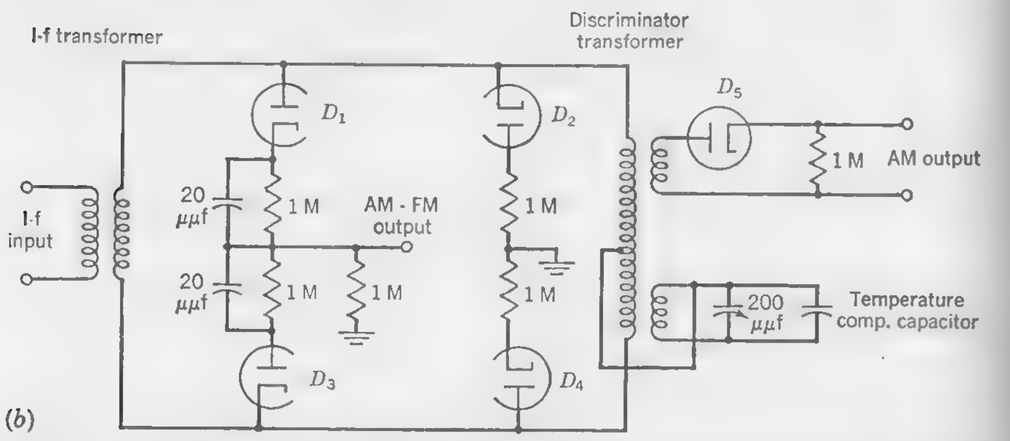


FIG. 3.2.17. Block diagram of an AM-FM multiplier (*a*), and an AM-FM detector-discriminator (*b*) especially designed for analog multiplication. The GAIN, FREQUENCY, and BALANCE controls are adjusted for zero output with $X_1 = 0$, $X_2 = 0$, and $X_1 = X_2 = 0$, respectively. The most sensitive parameter not compensated for through the use of similar networks is the center frequency of the discriminators, which may drift because of changes in secondary inductance, capacitance, or loading with temperature. Accuracies of 0.05 per cent of full scale, with response up to 20 kc, may be expected. In Fig. 3.2.17*b*, use of an intermediate-frequency transformer, typically 1.5 Mc, improves accuracy through filtering and permits use of peak detection to reduce loading effects which may make discriminator tuning a function of amplitude. Putting diodes in the primary of the discriminator transformer further reduces detuning effects due to changes in diode impedances with amplitude. Since outputs are connected to summing junctions of operational amplifiers, they are at ground potential; the AM output current is essentially independent of frequency. Diodes D_2 and D_4 serve to balance the discriminator circuit. (*Computer Systems, Inc.*)

The key elements of the time-division multiplier are accurate *electronic switches* which may be either voltage switches (Fig. 3.2.19*a*) or current switches (Fig. 3.2.19*b*,*c*) Figures 3.2.20 to 3.2.22 show complete time-division multipliers of several types.

3.2.13. Ripple Filtering vs. Frequency Response in Modulation-type Multipliers.[32] If, as is often the case, the multiplier output voltage is to be integrated, then little or no filtering of the pulsed-attenuator ripple is required. If, on the other hand, the multiplier feeds nonlinear devices, and, in particular, other similar multipliers, then one must reduce the rms ripple voltage below about 0.1 per cent of full scale to avoid

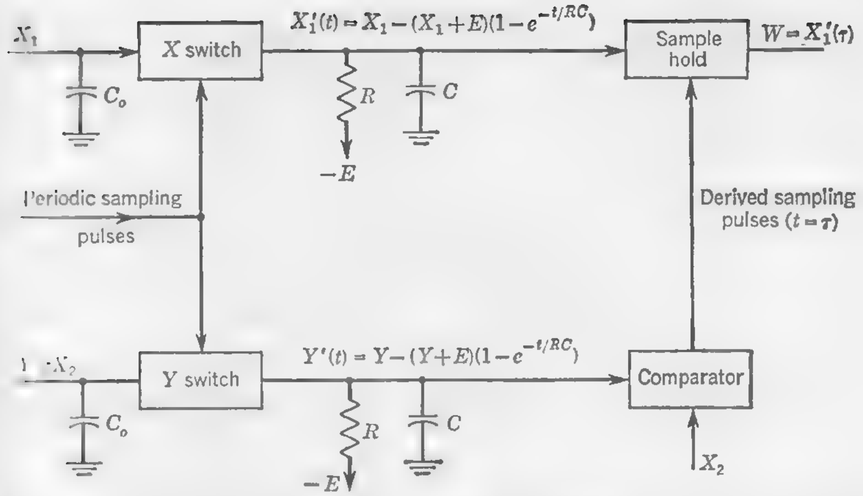


FIG. 3.2.18. Block diagram of a sampling multiplier. Capacitors C_0 prevent changes in X_1 and Y' during the short sampling interval during which time X_1' and Y' rise to X_1 and Y, respectively; X_1' and Y' then change in the manner indicated in the figure. $Y(>X_2)$ is usually a constant. When Y and X_2 are equal, the comparator emits a sampling pulse, so that the output W of the sample-hold circuit is

$$W = X_1'(\tau) = X_1 - (X_1 + E)\left[1 - \exp\left(-\log_e \frac{X_2 + E}{Y + E}\right)\right]$$
$$= \frac{X_1X_2 + EX_1 + EX_2 - YE}{Y + E}$$

The undesired terms are easily subtracted out. Accuracy to about 1 per cent at a 2- to 1 sampling rate may be expected. Substitution of different RC values in one of the two channels generates products of the form $X_1X_2^{\alpha}$.

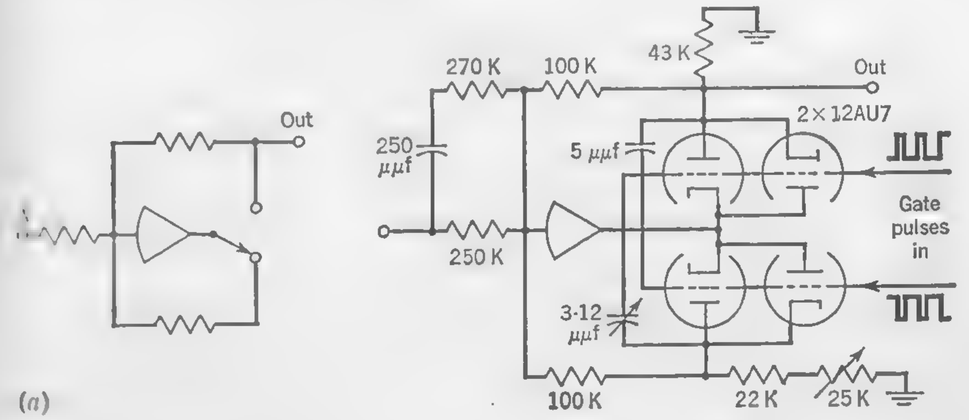
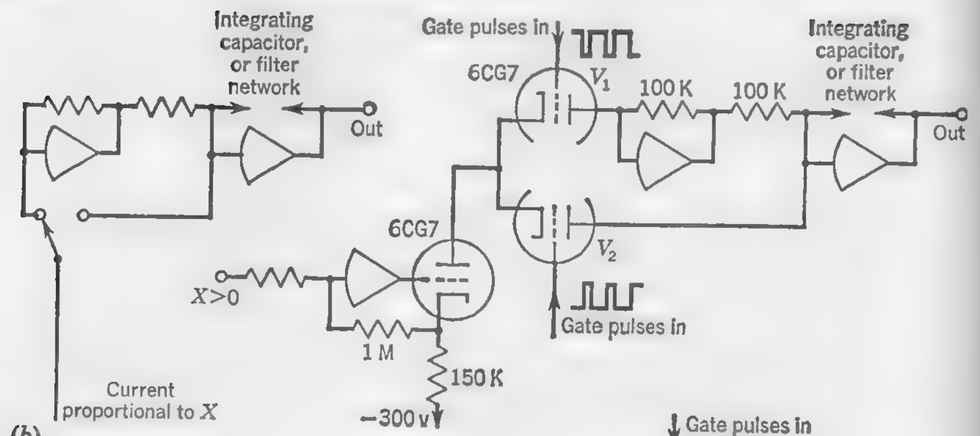


FIG. 3.2.19*a*. Precision bipolar electronic voltage switch. Amplifier is either chopper- or commutator-stabilized and, for stable, drift-free operation, must have good bandwidth relative to keying frequency. The RC series impedance at the input provides preemphasis to the input signal to compensate partly for the smoothing filter following the switch (not shown). The small capacitors are added to equalize signal injection from keying waveform (gate pulses) through interelectrode capacitances of switch tubes (12AU7). The variable resistor equalizes the time constant of direct loads on the switch (43 kilohms, 22 kilohms, and the potentiometer in series) to make recovery of each output during OFF cycle identical. Although not nearly so critical as the 100- and 250-kilohm resistors, the stability of the 43-, 22-, and 25-kilohm resistors is important to stable switch operation. (*Goodyear Aircraft Corporation.*)



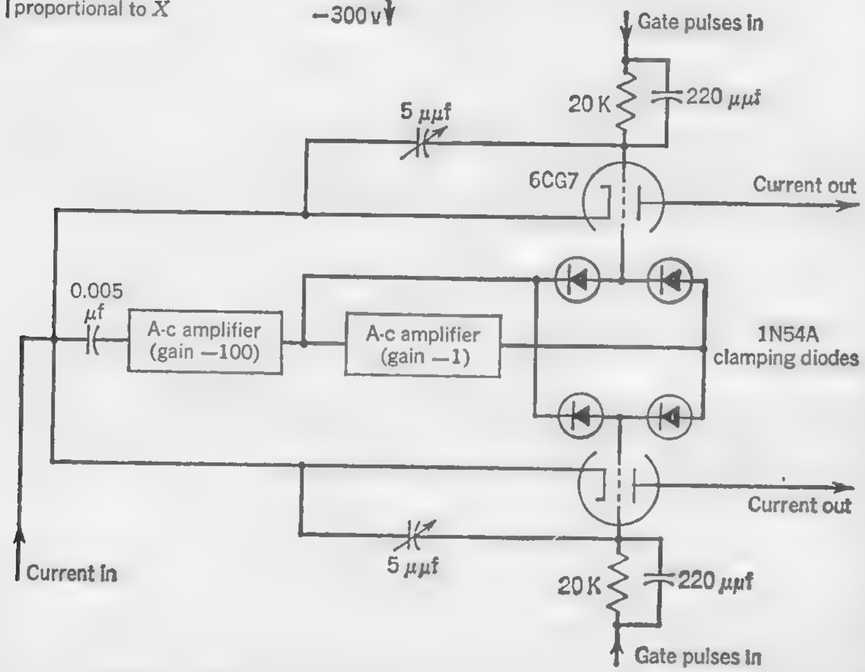
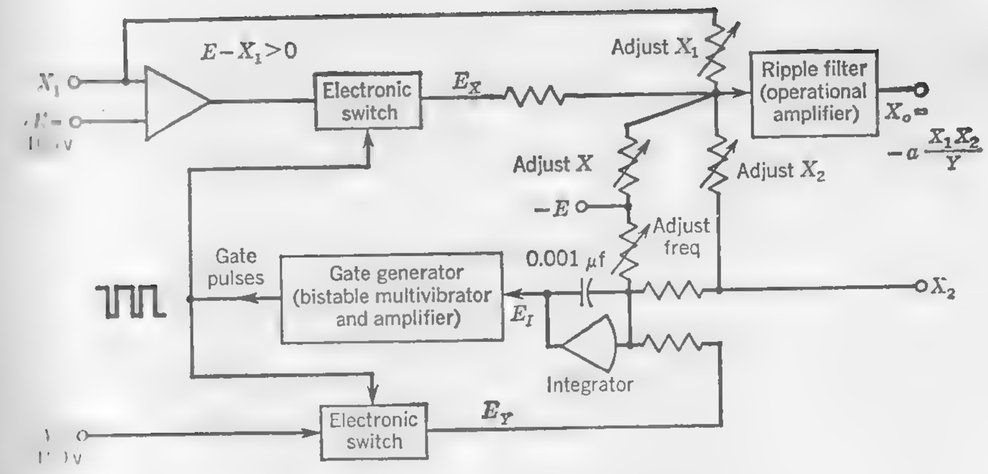


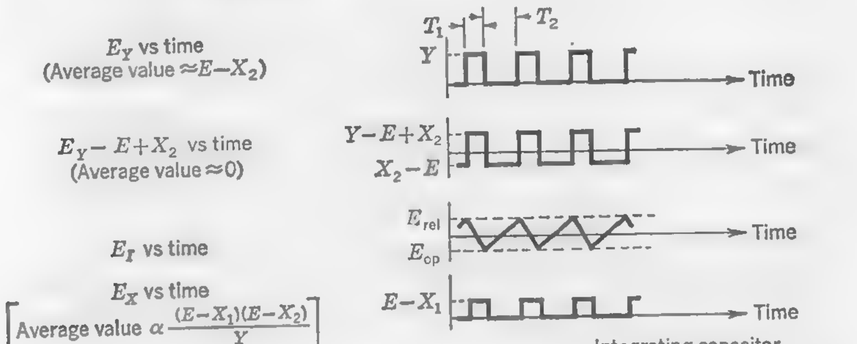
FIG. 3.2.19*b*. Precision monopolar electronic current switch. Amplifier should be chopper- or commutator-stabilized. Stability of all resistors shown is critical. Error due to current drawn by stray capacitance to ground at the cathodes of V_1 and V_2, and due to current injection from keying waveforms through the grid-cathode capacitances of V_1 and V_2 can be compensated in the manner of Fig. 3.2.19*c*.

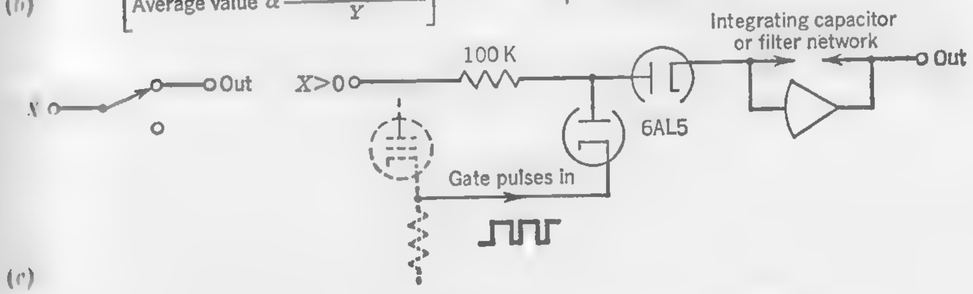
FIG. 3.2.19*c*. Dynamic compensation of a current switch. The first a-c amplifier (gain = −100) provides positive clamping voltages for the gate pulses to compensate for current losses through stray capacitance from the switch-tube cathodes (high-impedance output of current generator) to ground. The second amplifier provides negative clamping voltages to compensate for effects of unequal current injection through grid-cathode capacitances. (*RCA Laboratories Division.*)

serious errors due to ripple rectification. Since the two pulsed attenuators should be identical, the impedance presented by the output amplifier and/or ripple filter to the X_1 switch may have to be made equal to that of the amplifier, integrating circuit, or ripple filter connected to the Y switch.

The frequency response of a pulsed-attenuator multiplier is essentially determined by that of its ripple filter. For an rms ripple output of 0.1 per cent of full scale, the frequency response can be made flat within 0.5 db, and the phase shift kept below 3 deg, up to a frequency of the order of 1 per cent of the lowest repetition rate. Unfortu-







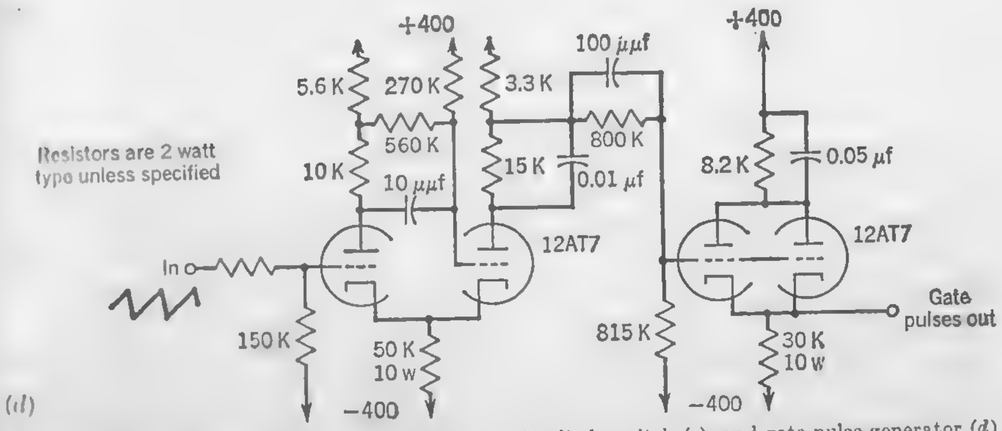


FIG. 3.2.20. Block diagram (*a*), idealized waveforms (*b*), diode switch (*c*), and gate-pulse generator (*d*) for a relatively simple time-division multiplier. The output voltage of the bistable multivibrator in Fig. 3.2.20*d* "flops" to its most negative value (and thus opens the switch) when the input voltage [illegible] below the hold value E_{op}; the output voltage "flips" back when the input voltage increases again to a second threshold level E_{rel}. Four-quadrant operation is obtained through addition of a voltage E to both X_1 and X_2 and subtraction of the unwanted product terms in the output amplifier. Accuracy is of the order of 0.3 volt with unstabilized d-c amplifiers, and the switching frequency is 40 to 50 kc. Chopper-stabilized amplifiers and a current-driven X_1 switch (see also Fig. 3.2.19) yield a static accuracy of about 0.1 volt. (*Donner Scientific Company; figure adapted from G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

nately, increased repetition rates imply less accurate electronic switching and increase stray-capacitance effects, so that one must make a compromise between static and dynamic accuracy. Complementary-waveform ripple injection (e.g., through the 1,200-$\mu\mu$f capacitors in Fig. 3.2.22) reduces noise without producing phase shift.

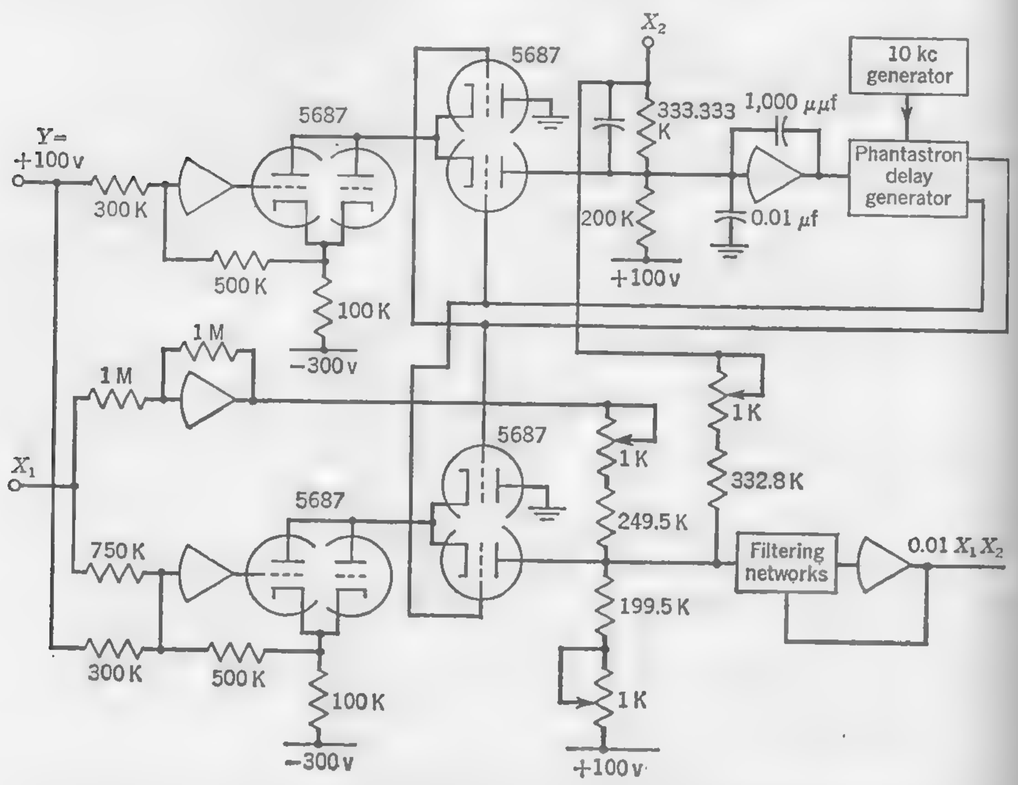


Fig. 3.2.21. A time-division multiplier using similar precision current switches. Not shown are a second slaved section producing XY_2, and switches and relays for making use of amplifiers when not used for multiplication, and to select desired filtering and dividing networks. The 10-kc generator is a conventional astable multivibrator. The delay generator is a five-tube circuit consisting of a phantastron, a squaring and phase-inverting circuit, and a pair of output circuits producing a rectangular waveform operating between −50 and −70 volts. Accuracy is within 0.01 to 0.04 per cent, depending partly on number of products (XY_n). With one filter setting, phase shift is as low as 0.5 deg at 100 cps. (*Electronic Associates, Inc.*)

ELECTRONIC DIVISION AND ELECTRONIC RESOLVERS

3.2.14. Electronic Division. Analog division by implicit computation (using high-gain amplifiers) is described in detail in Sec. 5.1. Many commercially built electronic multipliers have a built-in two-quadrant division mode which utilizes the output amplifier as a high-gain amplifier; as a rule some sort of high-frequency stabilizing network will be required (Fig. 3.2.23).

In feedback-regulated multipliers (Arts. 3.2.11 to 3.2.13) the reference voltage Y can, in principle, be varied to produce division by a variable Y. Since, however, changes in Y affect the loop gain and hence the accuracy and stability of the feedback regulation, this division method applies mainly to special-purpose computers where Y changes only over a narrow range, although automatic-gain-control circuits could conceivably be used (see also Sec. 7.1).

3.2.15. Electronic Resolvers and Inverse Resolvers (see also Secs. 3.1 and 7.1). In applications of electronic analog computers to aircraft and missile systems, the

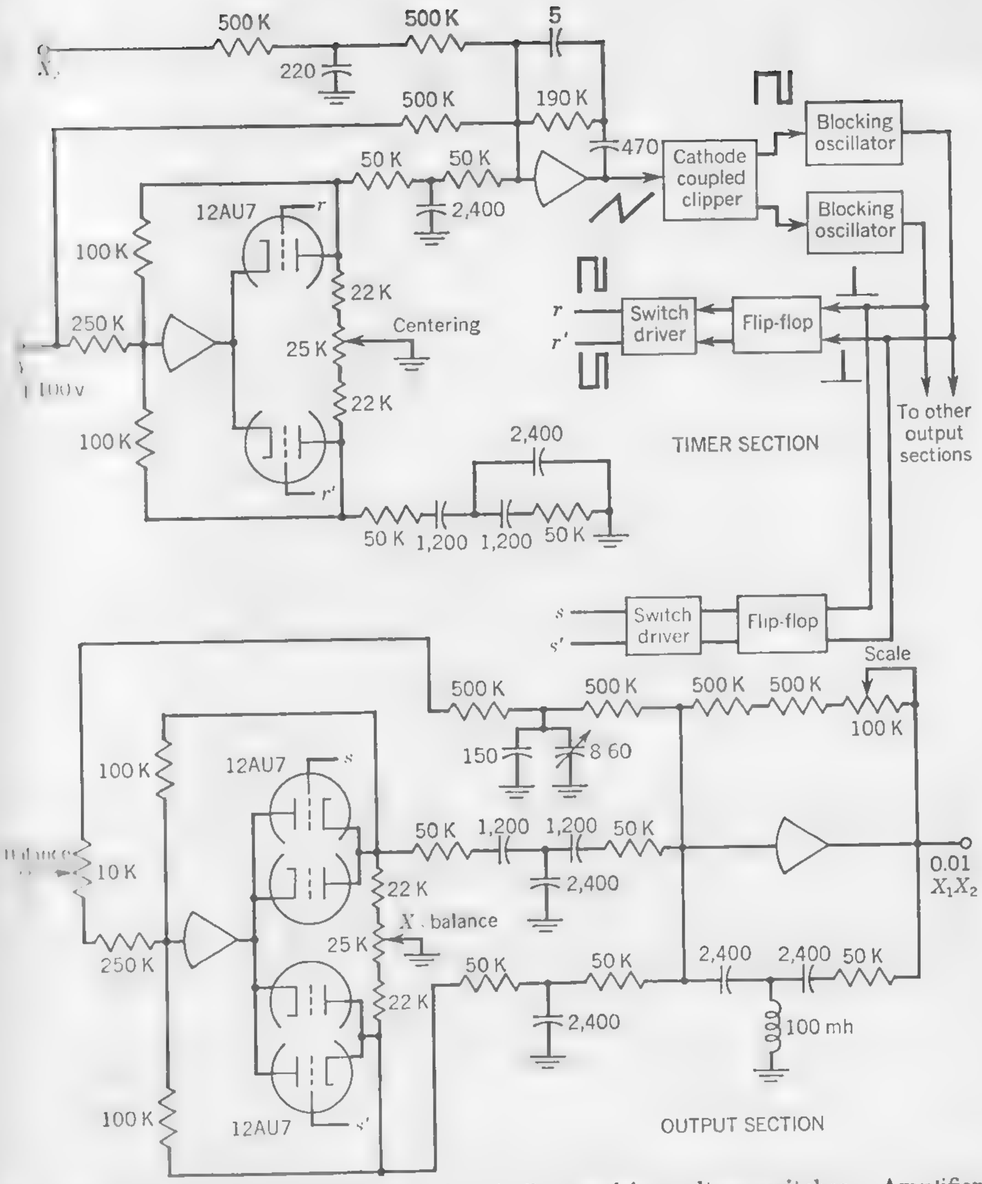


Fig. 3.2.22. A time-division multiplier using similar precision voltage switches. Amplifiers shown are stabilized wide-band d-c amplifiers. The cathode-coupled clipper, blocking oscillators, flip-flops, and switch driver are of conventional design; the switch driver is clamped on negative swings at approximately −100 volts using Zener diodes. Through the use of blocking oscillators, any reasonable number, usually one to five, of output sections may be slaved to a single timer without significant change in performance. The centering control enables one timer section to be matched to a number of output sections. X_2 balance matches each output section to the timer to which it is slaved. X_1 balance zeros the gain from the X_1 input to the output when $X_2 = 0$. The scale adjustment sets the output multiplying factor. The network between the off (upper) side of the X_1 switch and the output amplifier of the output section adds out-of-phase switch noise to greatly reduce over-all switching noise. An identical network in the timer section ensures that timer and output sections behave in the same way as X_2 is varied. Over-all accuracy is within 0.01 per cent of full scale. Phase shift is adjustable to zero at approximately 100 cps with a corner frequency of about 2,500 cps. (*Goodyear Aircraft Corporation.*)

accuracy and speed of resolvers and inverse resolvers are often the weak links in the computing system (see also Sec. 5.5).[30] Electronic resolvers can be more accurate as well as faster than servo resolvers at some expense in circuit complexity.

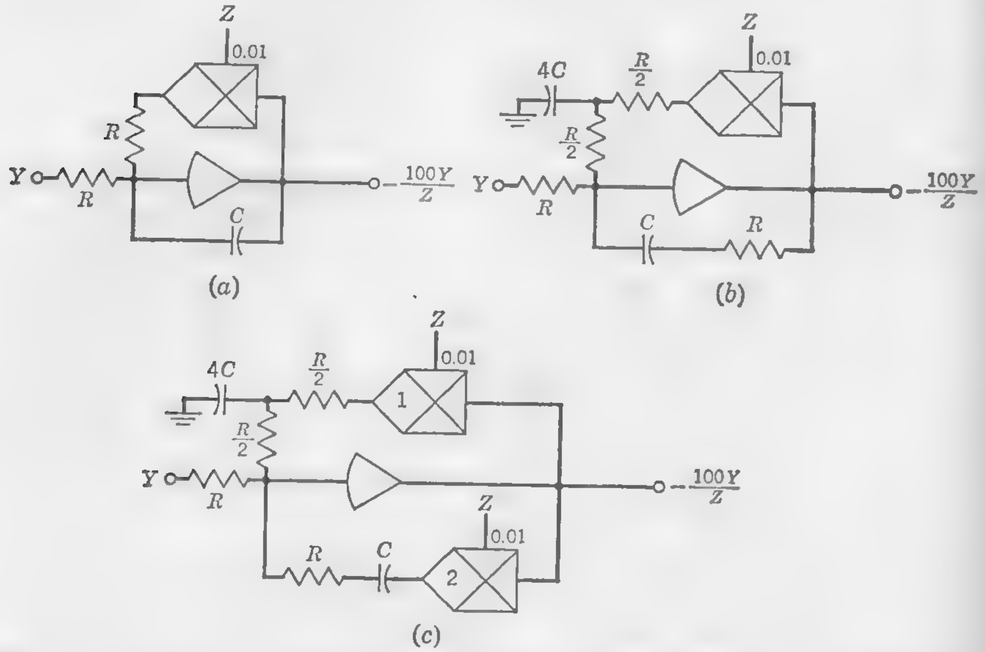


FIG. 3.2.23*a,b*. Electronic division using multipliers. Normal method (*a*) requires a capacitance in the amplifier feedback to prevent oscillations due to multiplier phase shift. Corner frequency of divider is at $\omega = (RCZ_{max}/Z)^{-1}$. The equally stable circuit (*b*) permits wide-band operation for large values of Z and lower phase shift for signal frequencies at all values of Z. $Z > 0$ in all cases.

FIG. 3.2.23*c*. Compensated electronic division. Multiplier 1 is a precision multiplier. Multiplier 2 is a relatively inaccurate wide-band multiplier, e.g., a simple quarter-square multiplier. Static accuracy is not improved with this method but phase shift is small for all values of Z. (*Goodyear Aircraft Corporation.*)

Electronic vector resolution requires implementation of machine equations like

$$X = R\cos\theta = R\cos\int_0^t \theta\,dt \qquad Y = R\sin\theta = R\sin\int_0^t \theta\,dt \tag{3.2.6}$$

or

$$X = U\cos\theta + V\sin\theta \qquad Y = -U\sin\theta + V\cos\theta \tag{3.2.7}$$

by one of the following methods:

1. *Straightforward combination of electronic multipliers and function generators.* Figure 3.2.24 shows a circuit employing quadrant switching of sine-cosine generators in the manner described in Art. 3.3.5.

2. Use of multipliers and integrators to generate sin θ and cos θ as *solutions of a differential equation* (Fig. 3.2.25). The input to such a device is $d\theta/dt = \theta$ rather than θ, and *sin θ and cos θ are produced without scale-factor restrictions on θ.*

3. *Voltage-controlled sampling of a sine-wave oscillator output* (see also Sec. 7.1).

4. *Digital-servo* generation of $R\cos\theta$ and $R\sin\theta$ (Secs. 3.3 and 7.1), or use of *digital function generators* (Sec. 3.3).

Methods 1, 2, and 4 permit static accuracies better than 0.1 volt and useful frequency response up to 100 cps at the expense of fairly complex circuitry.

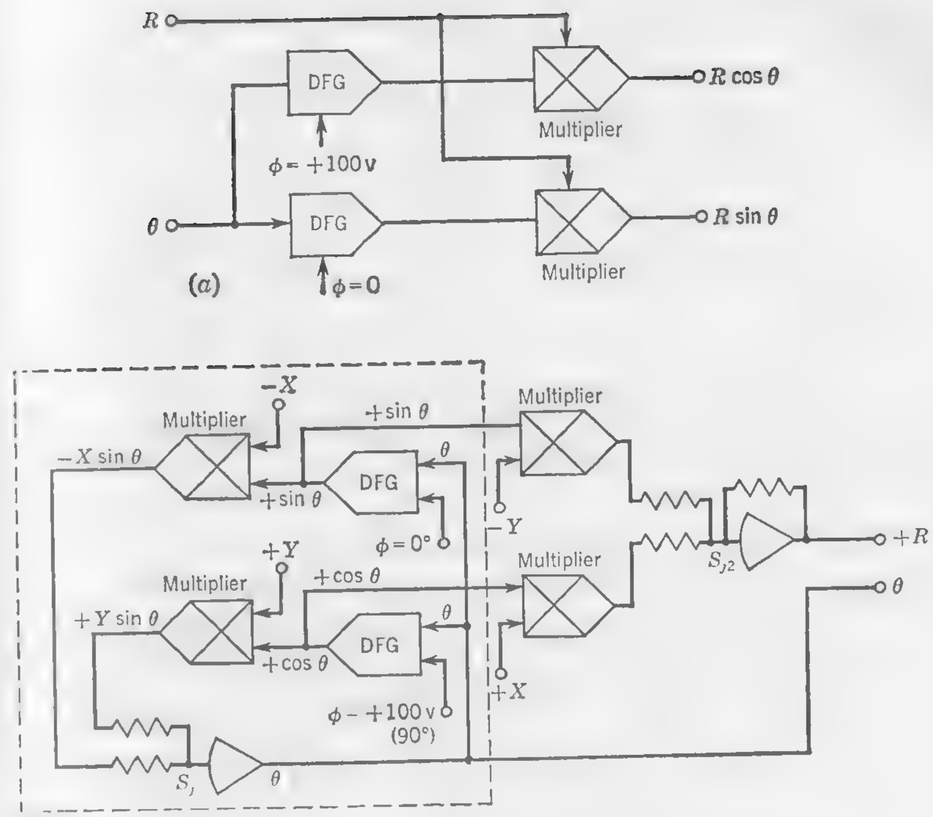


FIG. 3.2.24. Vector resolution (*a*) and composition (*b*) by means of electronic multipliers and diode function generators generating sin θ and cos θ. The sine-cosine generator shown in Fig. 3.2.24*c* (Beckman/Berkeley Type 1154) is an 18-segment diode function generator employing a 25-kc smoothing oscillator and −270-deg to +270-deg quadrant switching by means of a diode limiter (see also Sec. 3.3.5, Figs. 3.3.12 to 3.3.14). θ is scaled at 1.8 deg per volt. The resulting electronic resolver has a specified static accuracy better than 0.08 volt, less than 0.5 deg phase shift and 1 per cent distortion for 50 cps sine-wave output at 50 volts amplitude. A special patchbay layout permits separate use of multipliers and sine-cosine generators (Beckman/Berkeley Division).

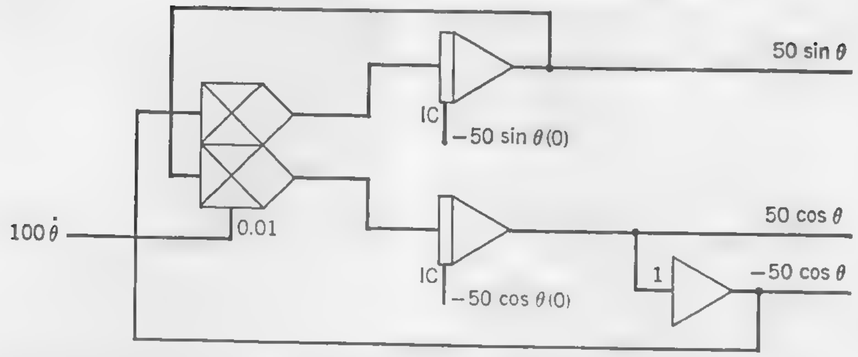


FIG. 3.2.25*a*. Electronic resolver using multipliers. This method provides good accuracy for short periods of time. When products involving sine and cosine are required, two separate multichannel multipliers are used.[30]

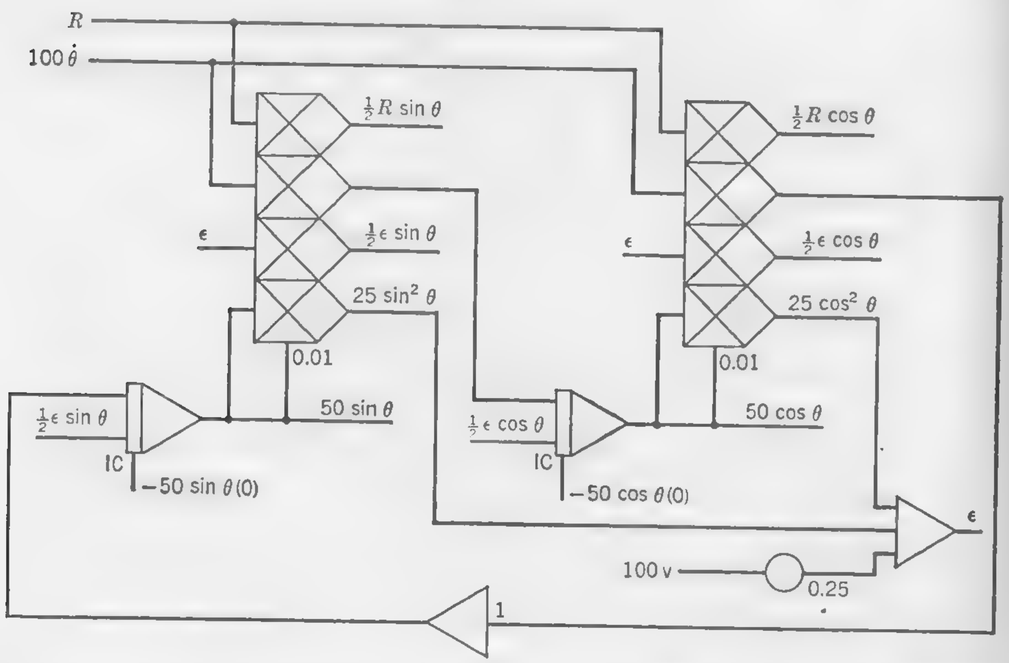


FIG. 3.2.25*b*. An improved version of Fig. 3.2.25*a*, in which $\sin^2\theta + \cos^2\theta$ is continuously compared with unity; the error is fed back to the integrators as a correction term.[30]

Inverse resolution or *vector composition* may involve direct computation of

$$R = \sqrt{X^2 + Y^2} \qquad \theta = \arctan \frac{Y}{X} \tag{3.2.8}$$

as shown in Sec. 7.1, or utilization of resolver circuits to generate R and θ by implicit solution of the equations

$$-X \sin\theta + Y \cos\theta = 0 \qquad X \cos\theta + Y \sin\theta = R \tag{3.2.9}$$

with the aid of a high-gain-amplifier loop with or without automatic gain control in the manner of Sec. 3.1. The vector-composition scheme of Fig. 3.2.26 combines the latter method with sine-cosine generation in the manner of Fig. 3.2.25.

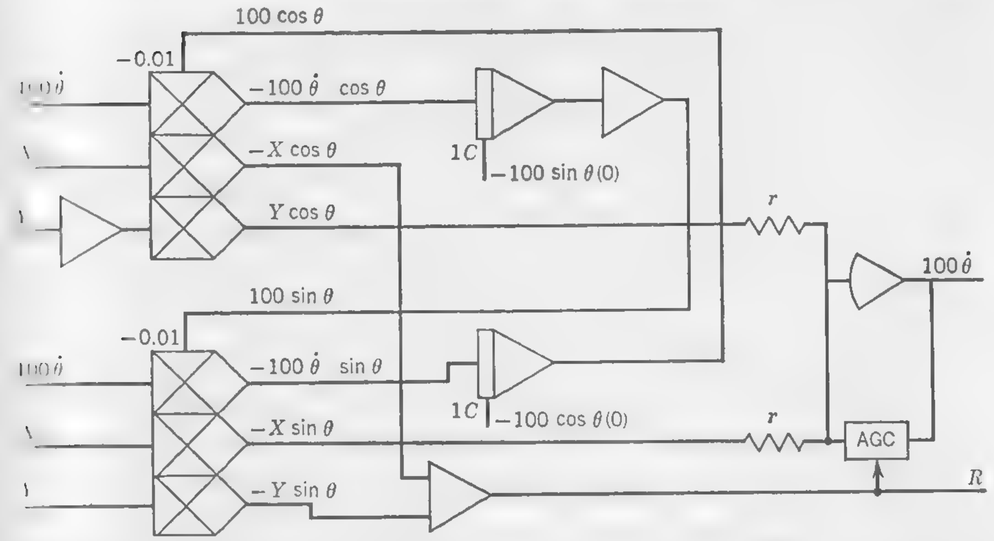


FIG. 3.2.26. Electronic vector composition. As in Fig. 3.2.25, an additional channel may be added to each multiplier to produce $\sin^2\theta + \cos^2\theta - 1$ as a correction signal. The AGC circuit, which is to keep the loop gain independent of R, may be a photoconductor and biased neon bulb.[7] Sometimes, for stability, a small capacitor must be used in the feedback path of the amplifier generating 100 $\dot{\theta}$.

References

1. Gilbert, R. W.: The Inductronic Product Resolver (IPR), Weston Electrical Instrument Corporation, *Prelim. Tech. Rept.* 356, Sept. 14, 1951.
2. Automatic Current Balance, Specialties, Incorporated, Descriptive Sheet S144A, December, 1950.
3. McNee, A. B.: An Electronic Differential Analyzer, *Proc. IRE*, **37**:1351–1324 (November, 1949).
4. Löfgren, L.: Analogue Multiplier Based on the Hall Effect, *J. Appl. Phys.*, **29**:158–166 (February, 1958).
5. Stevenson, E. C.: Application of the Gauss Effect in Bismuth to Analysis and Computer Circuitry, a monograph printed by the University of Virginia.
6. Hall, Albert C.: A Generalized Analog Computer for Flight Simulation, *AIEE Tech. Paper* 50-48, December, 1949.
7. Morrill, C. D., and R. C. Weyrick: An AGC System for D-C Servos, *Control Eng.*, July, 1958, p. 111.
8. SOLION Principles of Electrochemistry and Low Power Electrochemical Devices, U.S. Department of Commerce, Office of Technical Services, PB-131931, August, 1958.
9. Kovach, L. D.: An Analog Multiplier Using Thyrite, *IRE Trans. PGEC*, **EC-3**:42 (June, 1954).
10. D. G. O'Hare Company, *Monthly Progress Rept.*, Contract NOnr-1012(00), Dec. 10, 1952.
11. Soltes, A. S.: Beam Deflection Nonlinear Element, *Electronics*, **23**:122 (August, 1950).
12. Angelo, E. J., Jr.: An Electron Beam Tube for Analog Multiplication, *MIT Tech. Rept.* 249, Oct. 27, 1952.
13. Edwards, C. M.: Survey of Analog Multiplication Schemes, paper presented at ACM meeting, September, 1953.
14. Hardy, A. C., and E. C. Derch: An Electronic Method of Solving Simultaneous Equations, *J. Opt. Soc. Am.*, **38** (April, 1948).
15. Gray, T. S., and H. B. Frey: Acorn Diode Has Logarithmic Range of 10^9, *Rev. Sci. Instr.*, February, 1951, p. 117.
16. Stone, Joseph J.: Smooth Curve Function Generation Using Diodes, paper presented at Airborne Electronics Conference, May, 1954.
17. Meyers, R. A., and H. B. Davis: Triangular Wave Analog Multiplier, *Electronics*, **29**:182–185 (August, 1956).

18. Garrett, D. E., and F. G. Cole: A General Purpose Electronic Wattmeter, *Proc. IRE*, **40**:165–171 (February, 1952).
19. Mehron, M., and W. Otto: Instantaneous Multiplier for Computers, *Electronics*, **27**:144–148 (February, 1954).
20. Somerville, M. J.: An Electronic Multiplier, *Electronic Eng.*, **24**:78–80 (October, 1952).
21. Graig, L. J.: The Magnetic Amplifier as an Analog Computer Element, *Proc. IRE*, **41**:1477 (October, 1953).
22. Keister, G. L.: Transistor Magnetic Analogue Multiplier, *Electronics*, **29**:160–163 (October, 1956).
23. Baum, R. V., and C. D. Morrill: A Time Division Multiplier for a General Purpose Electronic Differential Analyzer, paper presented at the National Convention, IRE, 1951.
24. Chance, Britton, *et al.*: *Waveforms*, pp. 292–293, McGraw-Hill Book Company, Inc., New York, 1949.
25. Broomal, John, and Leon Riebman: A Sampling Analog Computer, *Proc. IRE*, **40**: 569–572 (May, 1952).
26. McCool, William A.: An AM-FM Electronic Analog Multiplier, *Proc. IRE*, **41**:1470–1477 (October, 1953).
27. Goldberg, E. A.: A High Accuracy Time Division Multiplier, *RCA Rev.*, **XIII**:265–274 (September, 1952).
28. Morrill, C. D., and R. V. Baum: A Stabilized Electronic Multiplier, *IRE Trans. PGEC*, December, 1952.
29. Steinberg, S.: An Accurate Electronic Multiplier, *RCA Rev.*, **16**:618–634 (December, 1955).
30. Howe, R. M., and E. G. Gilbert: Trigonometric Resolution by Means of Multiplier Elements, *IRE Trans. PGEC*, **EC-6**:(2) (1957).
31. Pfeiffer, P. E.: A Four-quadrant Multiplier, *IRE Trans. PGEC*, June, 1959.
32. Korn, Granino A., and T. M. Korn: *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.
33. Smith, G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1959.
34. Hartmann, J., G. A. Korn, and R. L. Maybach: Triangle-integration Multipliers, Department of Electrical Engineering, University of Arizona, *ACL Mem.* 18, 1960.

PART 3: ELECTRONIC FUNCTION GENERATORS, SWITCHING CIRCUITS, AND RANDOM-NOISE GENERATORS

By GRANINO A. KORN

DIODE LIMITERS AND SWITCHING CIRCUITS, OPERATIONAL RELAYS

3.3.1. Basic Diode Limiter Circuits.[1 to 4] The *basic diode limiter circuits* of Fig. 3.3.1*a* to *f* utilize the nonlinear current-voltage characteristics of vacuum or semiconductor diodes. The resulting input-output transfer characteristics can simulate important properties of dynamical systems (Secs. 5.3 and 5.4) or may be combined to produce a large class of arbitrary functions (Arts. 3.3.5 and 3.3.6).

Figure 3.3.1*g* to *m* shows *practical bias circuits* for various diode limiters used in combination with operational amplifiers. Finite diode forward and back resistances produce deviations from the ideal-diode characteristics shown (see also Art. 3.3.4). In the *feedback limiter* circuits of Fig. 3.3.1*k* to *m* the limiting portions of each characteristic are not ideally flat but have a finite slope equal to $-\rho R_0/R_1(\rho + R_0)(\approx -\rho/R_1$ if $\rho \ll R_0)$, where ρ is the sum of the diode forward resistance R_D and r, r_1, or r_2. For good limiting, one often makes E large (300 volts) and lets r and r' be as small as possible without undue loading of the amplifier.

In general, *series limiters* (Fig. 3.3.1*a,b,c,g,h,i*) limit more accurately than *shunt limiters* (Fig. 3.3.1*d,e,f*) and feedback limiters. The *dual series limiter* (Fig. 3.3.2) and the especially useful *bridge limiter* (Fig. 3.3.3) are series limiters whose transfer characteristics are similar to those of dual shunt or feedback limiters.

3.3.2. Precision Limiter Circuits.[1 to 3] While the limiter circuits of Figs. 3.3.1 to 3.3.3 may involve static errors between 0.2 and 1 volt, depending on the particular application, the *precision limiter circuits* of Fig. 3.3.4 produce static accuracies within 0.05 volt with the aid of chopper-stabilized high-gain d-c amplifiers. When each diode D_1 conducts, it is inside a high-gain degenerative feedback loop, so that diode forward resistance and "built-in bias voltages" are divided by the loop gain. As D_1 ceases to conduct, the open-loop gain of the amplifier produces a sharp step at X_0' and cuts D_1 off decisively. Figure 3.3.5 shows some actual limiter characteristics.

Figure 3.3.6 shows dual-limiter and dead-space circuits using a single amplifier to produce precision limiter action. With suitable amplifiers and diodes (Art. 3.3.4), such circuits can be used even at high audio frequencies in connection with triangle-integration multipliers (Sec. 3.2) and special switching applications.

3.3.3. Comparators, Multivibrators, and Other Switching Circuits (see also Secs. 4.3, 5.4, and 6.3).[1 to 9] Diode circuits can give an analog computer much of the decision making and automatic-programming capability of a digital computer. Figures 3.3.7 to 3.3.11* show a number of useful circuits. In particular, the *comparator* circuit of Fig. 3.3.8 produces decisive voltage steps whenever the sum of its input voltages changes sign and can drive *operational relays* to change programs, generate functions, etc. (see Secs. 4.2 and 5.3 for applications). Polarized relays may also be driven by special differential amplifiers, but the use of the comparator circuit does away with the need for polarized relays and special components.

With faster computing speeds, relay operating times (0.5 to 2 msec) introduce errors, and relays are replaced by the accurate *electronic switches* described in Sec. 6.3. In particular, the diode-bridge limiters of Figs. 3.3.3 and 3.3.6 can be operated as accurate electronic switches through reversal of their bias voltages. In the circuits of Fig. 3.3.6 the switch is inside a feedback loop, so that the forward resistance of the switch is effectively divided by the loop gain.

Electronic switches, like relays, can be operated by comparator or multivibrator output steps (see Secs. 5.4 and 6.3 for applications). Note also that *every diode circuit operating precision limiter action* (*Figs.* 3.3.4, 3.3.6, 3.3.7*b*, 3.3.10, 3.3.11) *supplies comparator steps in addition to its limiter output* and can, thus, operate relays, lights, or electronic switches indicating state changes.

3.3.4. Diode Specifications. High-frequency Behavior of Diode Circuits. The following tabulation compares the characteristics of actual vacuum and solid-state diodes with those of an ideal diode. All diodes should withstand between 100 and 200 volts peak inverse voltage.

	Forward resistance, ohms	Back resistance, at 10 volts, megohms	Capacitance at 10 volts reverse, $\mu\mu$f
Ideal diode	0	∞	0
Vacuum diodes (e.g., ½ 6AL5)	200–1,500	50–700	2–10
Silicon-junction diodes (e.g., 1N302A)	5 – 50	50–200	10–50
High-speed silicon-junction diodes (e.g., 1N643)	100 – 300	0.1–100	1– 8
Germanium point-contact diodes (e.g., 1N39B)	10 – 100	0.1–0.5	1– 5

In addition, vacuum tubes have possibly objectionable capacitances to ground. The effective zero-bias breakpoints of vacuum diodes are slightly negative, and those of solid-state diodes are slightly positive ("built-in bias" of 0.1 to 0.6 volt). In critical applications breakpoints must be stabilized through filament-voltage regulation (vacuum diodes), with thermistors in the bias networks (solid-state diodes), or with com-

* The writer and publishers are grateful to Donner Scientific Co., Concord, Calif., and to D. Taskett, Director of Sales, for their permission to reproduce Fig. 3.3.11 from G. A. Korn, *Nonlinear Operational-amplifier Circuits for Storing Maxima and Minima*, Memorandum dated Feb. 12, 1959. These circuits were developed to mark and store the maximum acceleration in a Donner accelerometer application.

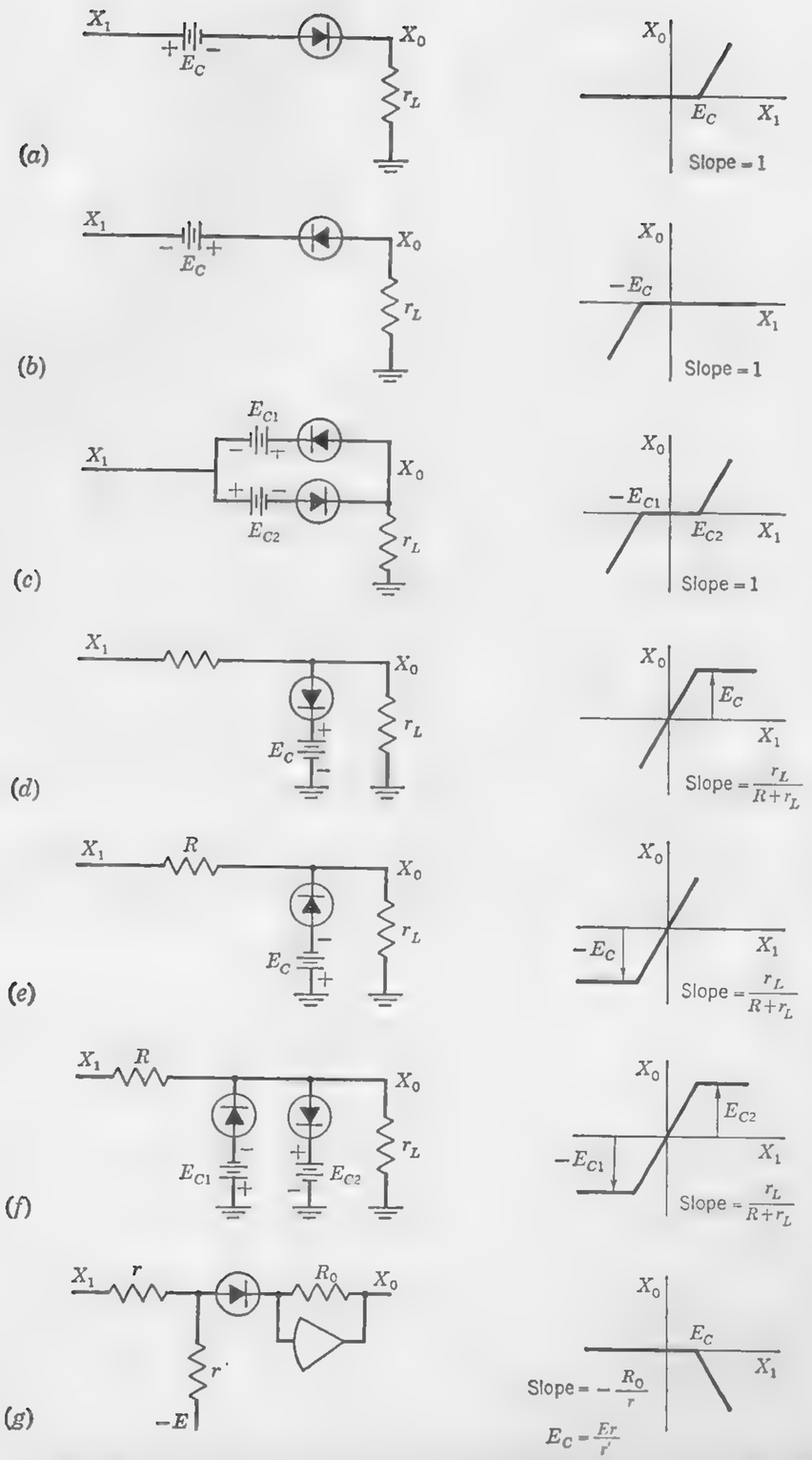
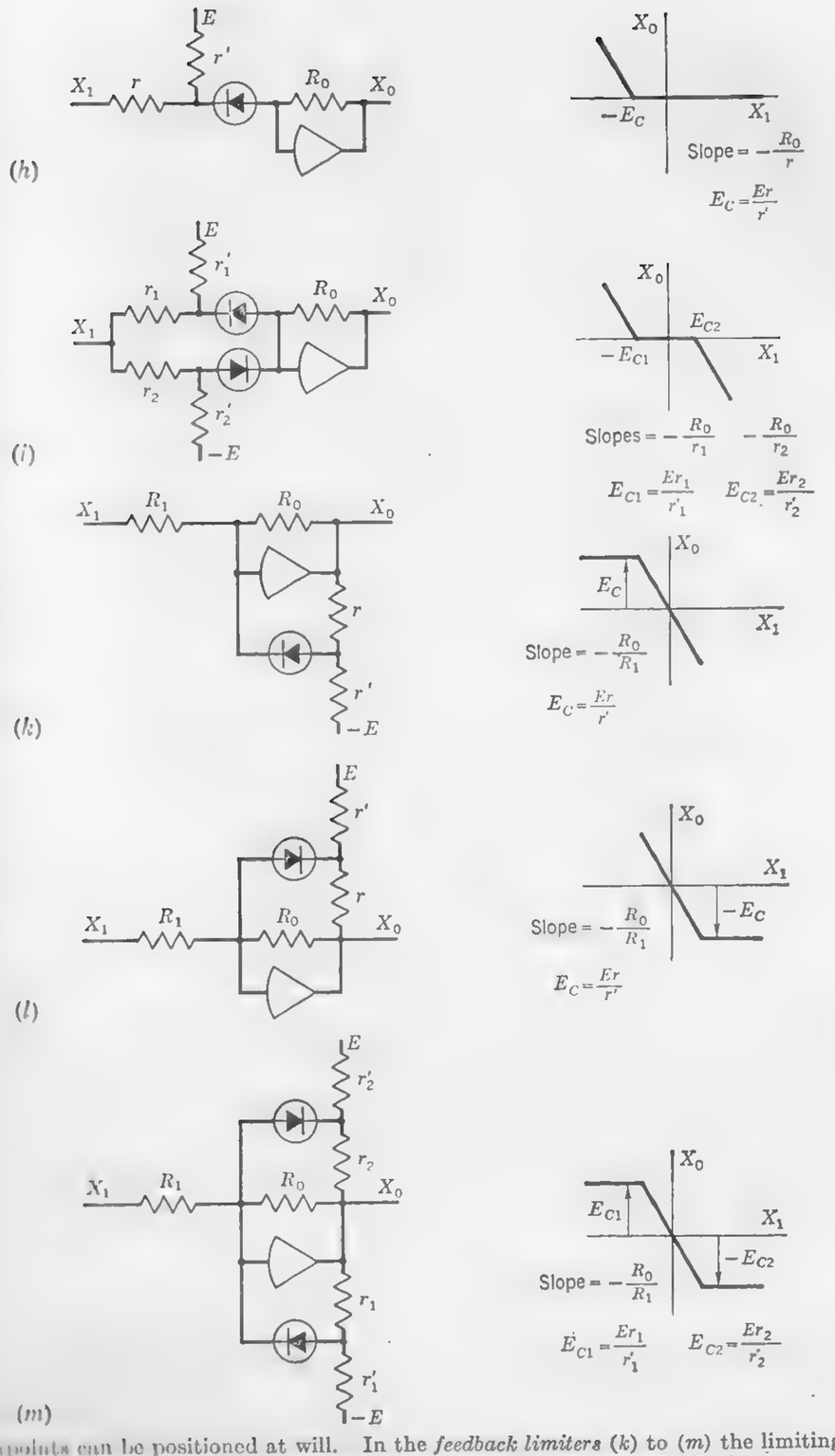


FIG. 3.3.1. Basic *series diode-limiter circuits* (*a*) to (*c*), *shunt limiters* (*d*) to (*f*), and practical operational-amplifier circuits (*g*) to (*m*). The transfer characteristics shown assume ideal diodes and zero-impedance voltage sources. Note that E_c may be negative as well as positive, and E_{c1} and $-E_{c2}$ can both be positive or negative, as long as $E_{c1} > -E_{c2}$; thus



breakpoints can be positioned at will. In the *feedback limiters* (*k*) to (*m*) the limiting portions of each characteristic actually have finite slopes given by $-rR_0/R_1(R_0 + r)$, so that one requires small values of r for accurate limiting.

pensating diodes (two-diode circuits like that of Fig. 3.3.7*a* reduce the effective breakpoint drift by a factor of 2 to 5).

Circuit resistances associated with diode limiters should be designed so as to "swamp out" diode back or forward resistances, whichever is more important. The effects of finite back resistance (leakage) are reduced by circuits which combine series and shunt limiting (Figs. 3.3.4, 3.3.7, and 3.3.10). Precision-limiter circuits (Figs. 3.3.4 and 3.3.6) eliminate the effects of breakpoint drift and forward resistance at low frequencies. They also reduce the effects of finite back resistance, especially if combined series and shunt limiting is used (Fig. 3.3.7).

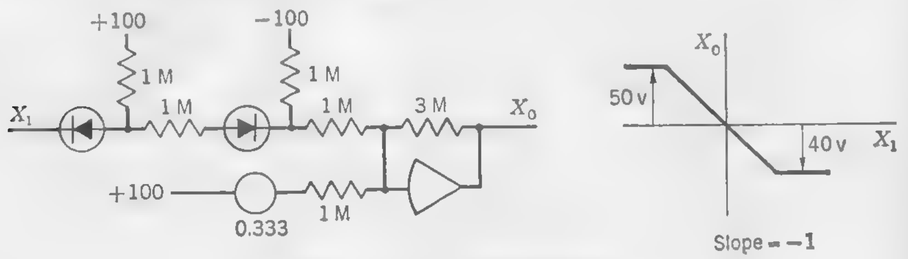


FIG. 3.3.2. Dual series limiter.

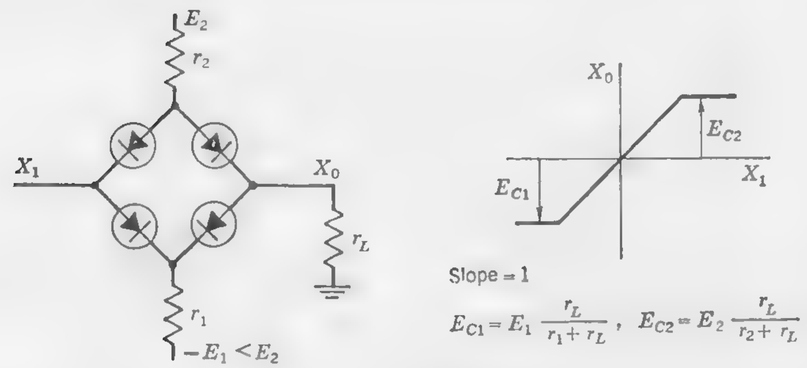


FIG. 3.3.3. Bridge limiter.

The effects of diode and cabling capacitances on diode limiter performance become noticeable at frequencies as low as 2 cps and cause hysteresis-like "spreading" of the transfer characteristics for increasing and decreasing voltages, capacitive leakage through limiters, and integration in circuits comprising feedback diodes. Capacitive as well as resistive leakage is minimized by combined shunt and series limiting. Carefully designed diode function generators permit 0.5-volt accuracy beyond 100 cps. With short leads, low-capacitance diodes, and suitable amplifiers, the precision-limiter circuit of Fig. 3.3.4 will limit with an accuracy of 1 per cent of peak output at frequencies exceeding 50 kc.

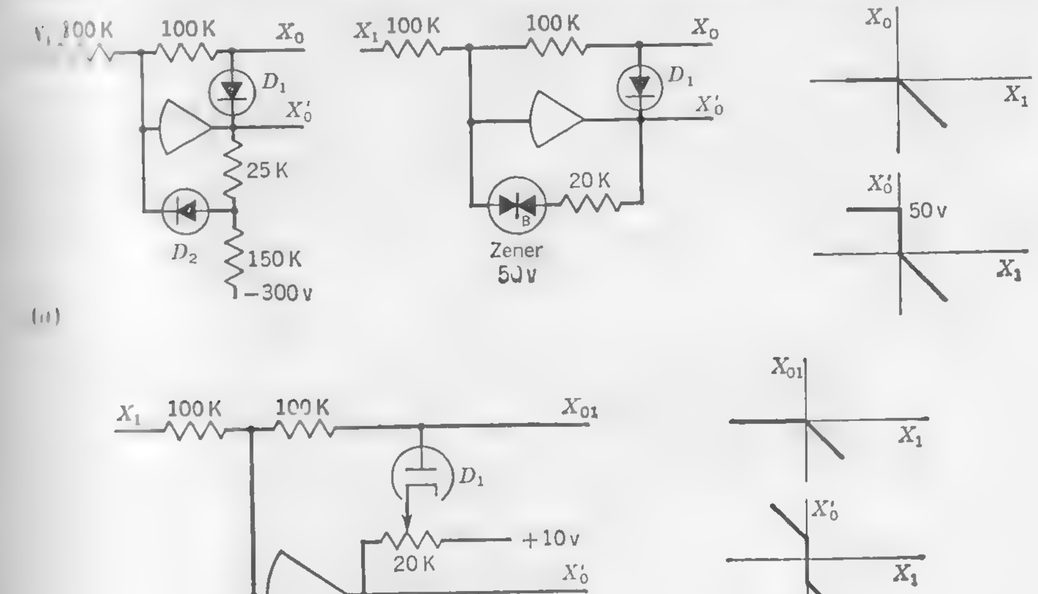
DIODE FUNCTION GENERATORS

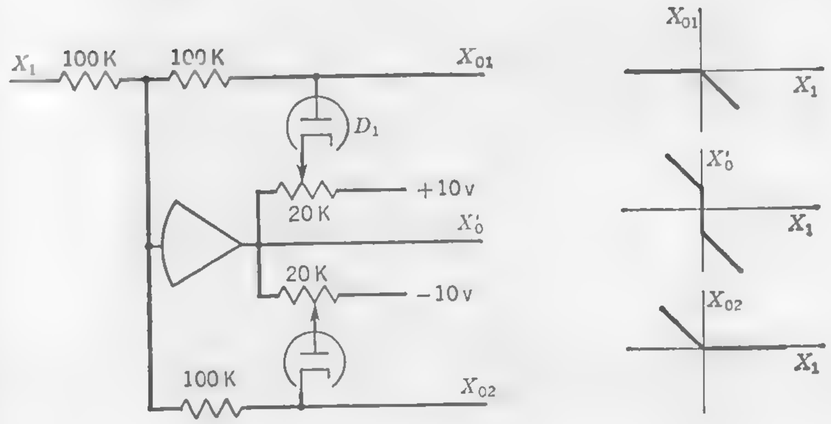
3.3.5. Special-purpose Function Generators. Use of Smoothing Oscillators. The broken-line transfer characteristics of *resistive networks containing biased diodes* can approximate a large class of functions; Fig. 3.3.12 shows three typical arrangements. Nonmonotonic functions must be obtained by subtraction of a nondecreasing function and a nonincreasing function. *For best accuracy the diodes should be used only to correct a linear or analytic approximation to the desired function.*

Figure 3.3.13 illustrates the use of a high-frequency (5 to 500 kc) *smoothing oscillator* for rounding diode-function-generator characteristics near the break points.[3] The triangular or sinusoidal smoothing-oscillator voltage is added to the function-generator

input, which is thus swept across the breakpoints; an output-averaging filter yields the smoothed characteristic. The same principle is used in triangle-integration squarers and multipliers (Sec. 3.2).

Figure 3.3.14*a* shows the use of a diode limiter to produce the *absolute value* of an input X_1. Such circuits are used in instrument inputs and to generate functions





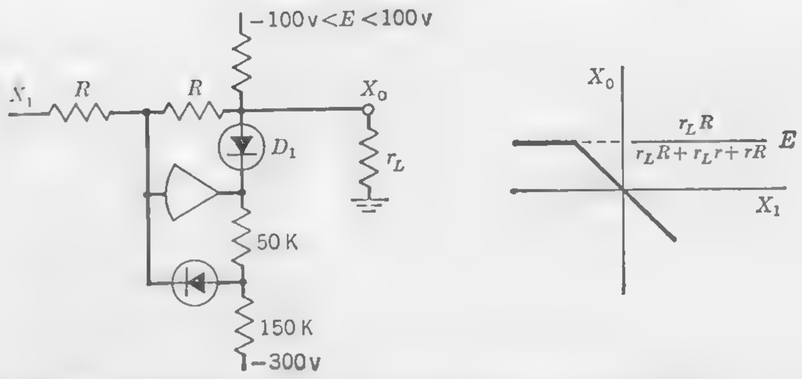


FIG. 3.3.4. Precision-limiter circuits. The biasing circuit in Fig. 3.3.4*b* can be omitted if silicon-junction diodes are used, since such diodes have "built-in bias."[1,5]

accurately symmetrical about $X_1 = 0$. Figure 3.3.14*b* and *c* shows the analogous use of dual limiters and dead-space circuits as *quadrant switches* for sine and cosine generators. *The precision-limiter circuits of Figs.* 3.3.4 *and* 3.3.6 *are recommended for these applications.* They not only permit static accuracies better than 0.05 volt, but their built-in comparator outputs may be used to light polarity- or quadrant-indicating neon lights, if desired.

3.3.6. Diode Universal Function Generators.[2,3] The diode function generator of Fig. 3.3.15*a* permits one to approximate a wide variety of functions systematically by summing the output voltages of simple series-limiter channels. Figure 3.3.15*b* shows another diode function generator based on shunt-limiter circuits.

3.3.7. Function Setting and Storage.[2,3] One determines break points and slopes for a given function $X_0 = F(X_1)$ by fitting a graph of the function (or of the appropriate *correction function*, Sec. 5.1) with the desired number of straight-line segments. For special-purpose function generators, the required resistance values can then be computed on the basis of ideal diode characteristics. Final adjustments should be made empirically, with the function-generator output connected to a servo plotting board, to a comparison potentiometer, or to a servo or digital voltmeter.

In the universal function-generator circuits of Fig. 3.3.15, each limiter-channel output affects only function values corresponding to arguments X_1 beyond its break point

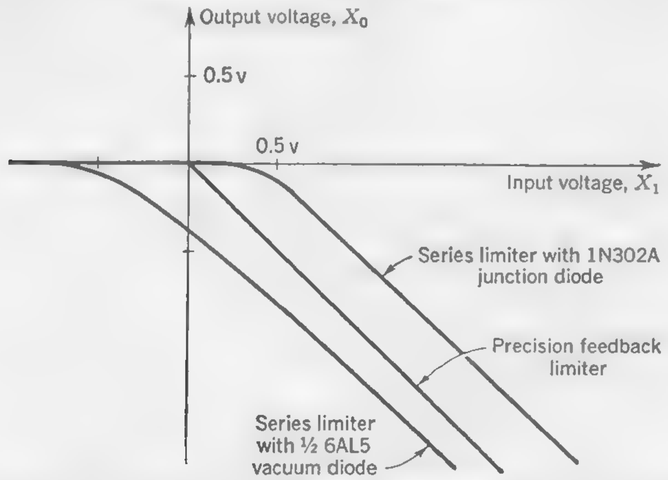


FIG. 3.3.5. Reproduction of actual d-c transfer characteristics of the series-limiter circuit of Fig. 3.3.1*g* (followed by a phase-inverting amplifier) with vacuum and solid-state diodes, and of the precision limiter of Fig. 3.3.4*a* ($E_c = 0$).

Break-point and slope settings can be made by reference to *calibration curves* based on actual diode characteristics; instead, one usually sets all slopes initially to zero and adjusts breakpoints and slopes empirically for successively higher values of X_1. The entire procedure should be repeated once or twice in order to allow for possible interference between breakpoint and slope adjustments.

Ease of *function changing and storage* (and hence resettability) are design features of great importance in multipurpose computers. In large computers, slope and breakpoint adjustments are set by *automatic potentiometer-setting systems* (Sec. 2.1), so that settings can be stored on punched tape; *card-controlled diode function generators* are also commercially available. In medium-priced computers *storable plug-in function-setting networks* (possibly using fixed break points for simplicity) are probably the best solution.

Wherever accuracy is important, the outputs of arbitary-function generators must be used only as corrections to analytic approximations generated by linear computing elements and accurate multipliers: frequently linear approximations are sufficient (Sec. 5.1). In addition, every attempt should be made to utilize the full dynamic range of function-generator input and output variables through efficient scaling. In this manner accuracies better than 0.2 per cent can often be realized.

CATHODE-RAY-TUBE FUNCTION GENERATORS

3.3.8. Photoformers.[10 to 12] In the *photoformer* (Fig. 3.3.16*a*), a feedback loop comprising the cathode-ray tube, the photocell, and the vertical deflection amplifier keeps the spot on the screen just below the edge of a prepared *function mask* as the input

voltage X_1 (horizontal deflection voltage) varies. Hence the output voltage X_0 (vertical deflection voltage) varies as a function $X_0 = f(X_1)$ of X_1.

3.3.9. Cathode-ray-tube Scanners.[13,14] In a second type of cathode-ray-tube function generator, the electron beam is positioned horizontally by the input voltage X_1,

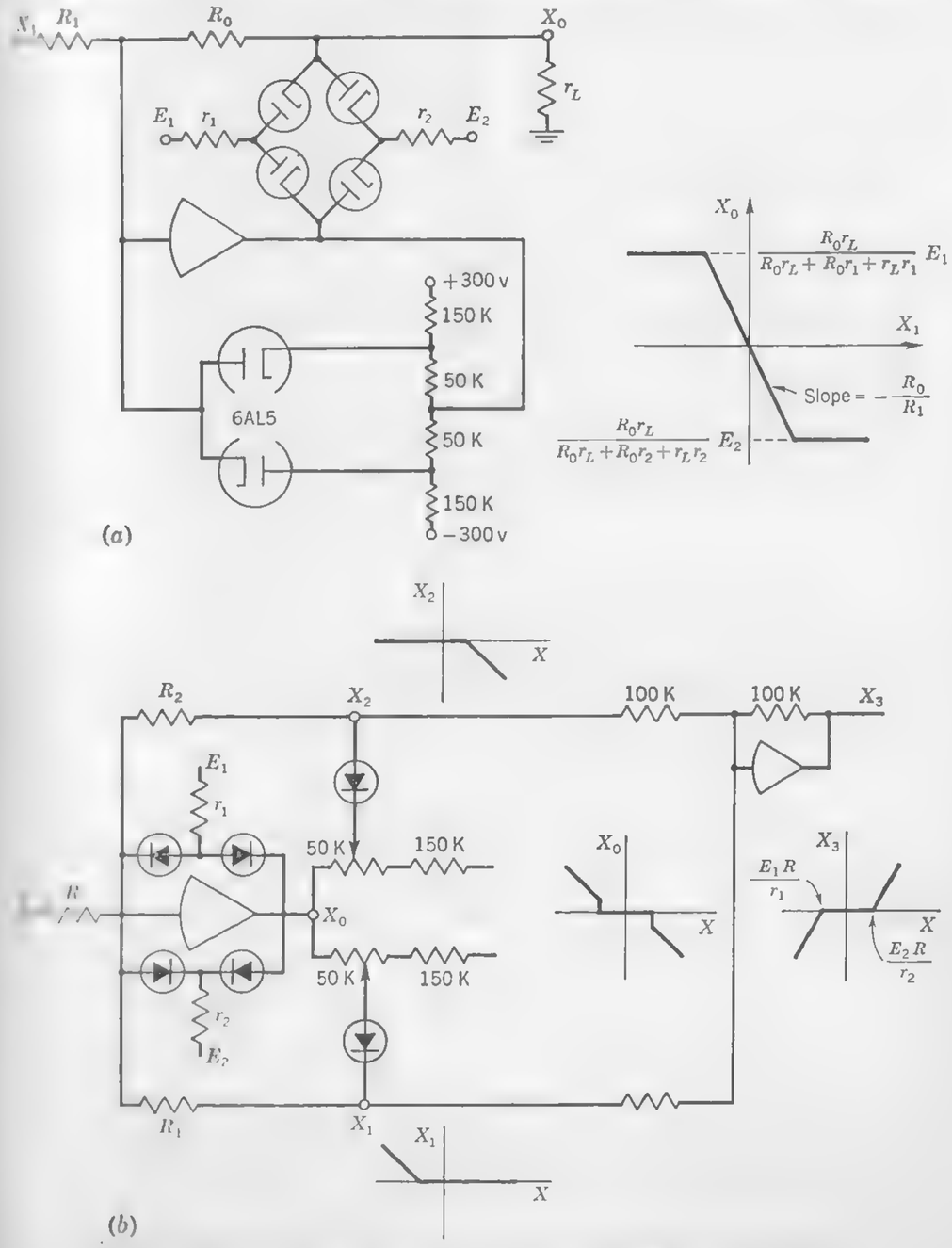


FIG. 3.3.6. Precision dual limiter (*a*), and precision dead-space circuit (*b*).[5,6]

while a linear sawtooth sweep causes the beam to sweep vertically so as to scan an opaque or transparent function mask between 5,000 and 50,000 times per second. For each value of X_1, the photocell current is, then, pulse-width-modulated in accordance with the function $F(X_1)$ used to shape the mask. The resulting pulse-width-

modulated square wave is filtered to yield $F(X_1)$ and could even be used to multiply other variables by $F(X_1)$, as in a time-division multiplier. Accuracy and frequency response are comparable with those of photoformers; the scanning scheme reduces some of the errors due to screen and phototube persistence, changes in spot size, and

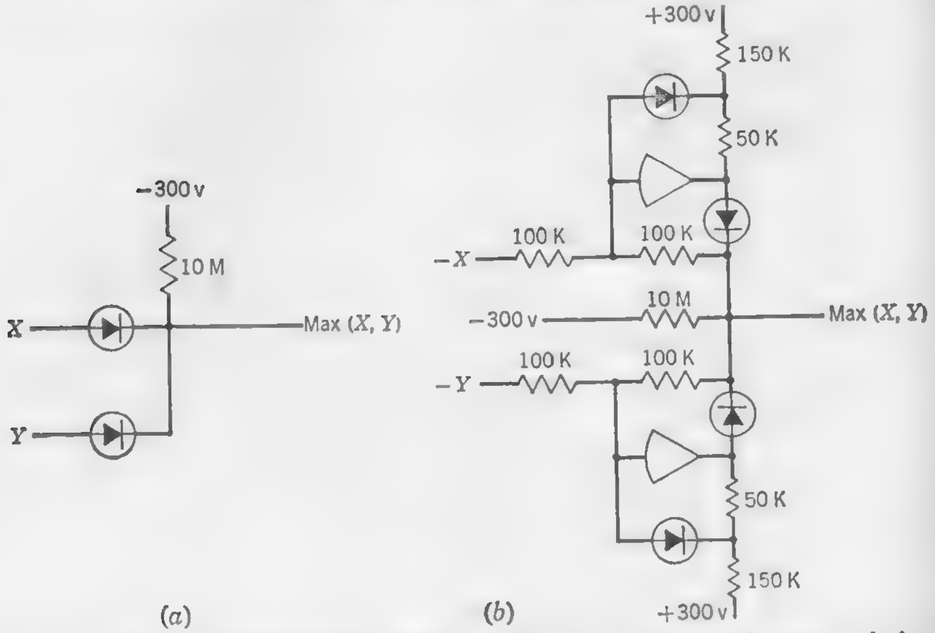


FIG. 3.3.7. These amplitude-selector circuits are useful as limiters and gates, and also for various logical operations. These symmetrical circuits have low output impedances in either state and tend to cancel break-point drift due to changes in temperature or diode cathode emission. The circuit of Fig. 3.3.7*b* incorporates precision-limiter action.[2,5]

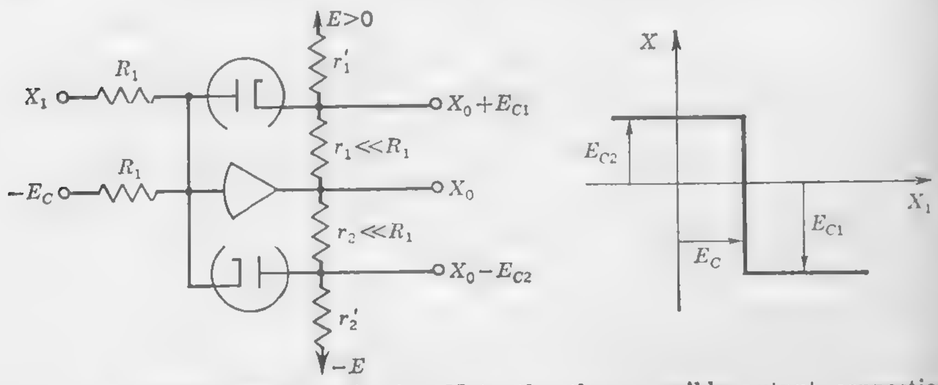


FIG. 3.3.8. Analog-comparator circuit. Note the three possible output connections. The circuit can drive relays, electronic switches, indicator lights, etc.; addition of a bridge limiter (Fig. 3.3.3) yields accurately limited output levels suitable for further computation.[1,5] $E_{c1} = Er_1/r_1'$, $E_{c2} = Er_2/r_2'$.

variations of deflection sensitivity. Provision of an opaque or transparent *reference line* yields standard reference pulses for automatic calibration.

3.3.10. Appraisal. Cathode-ray function generators permit convenient and rapid function changes if prepared function masks are available. Static accuracy is better than 0.5 per cent of full scale; response up to 100 kc is possible and permits application to repetitive analog computers (Sec. 6.3). One can improve the effective accuracy by

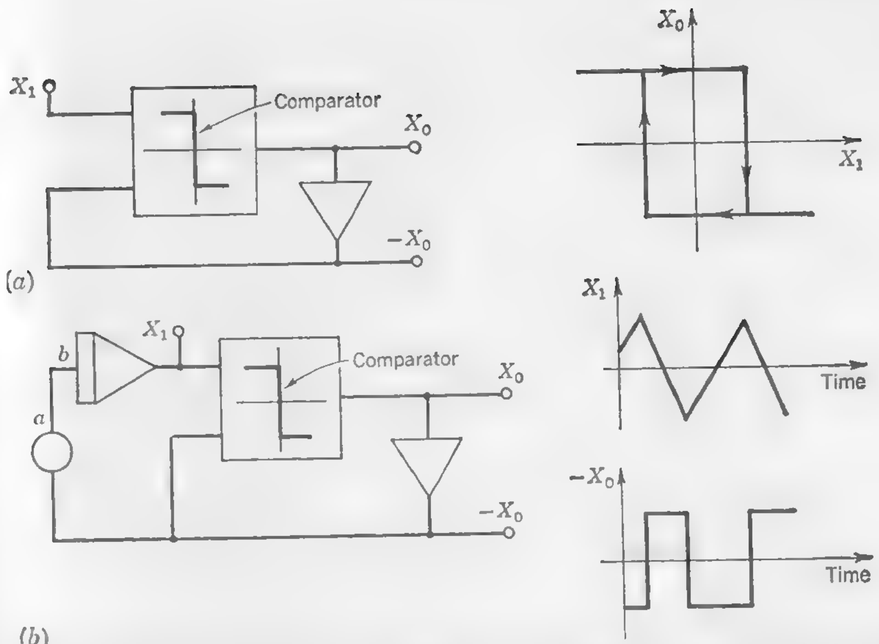


FIG. 3.3.9. Bistable multivibrator circuit with push-pull output. Addition of the integrator circuit shown in dash lines yields a free-running multivibrator which generates accurate triangle and pulse waveforms; both pulse-amplitude and pulse-width modulation may be produced.[1,5]

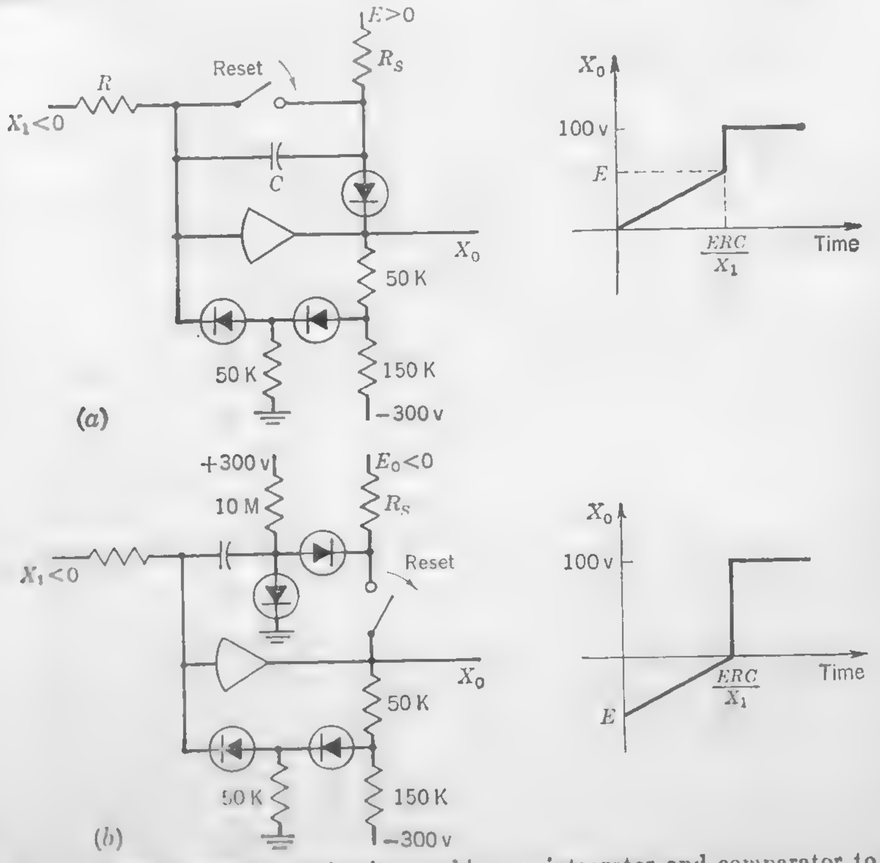


FIG. 3.3.10. These analog timer circuits combine an integrator and comparator to mark accurate time intervals. Note the low-leakage feedback limiter used.[5]

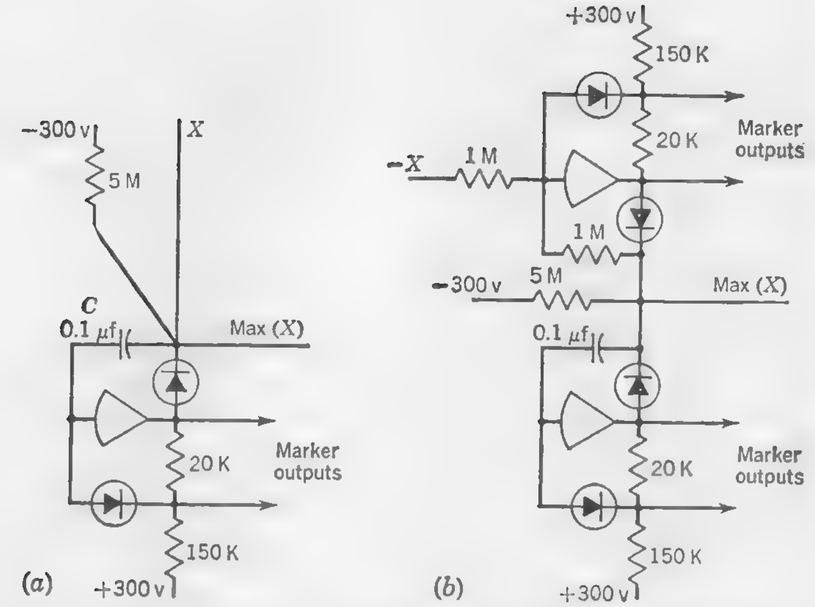
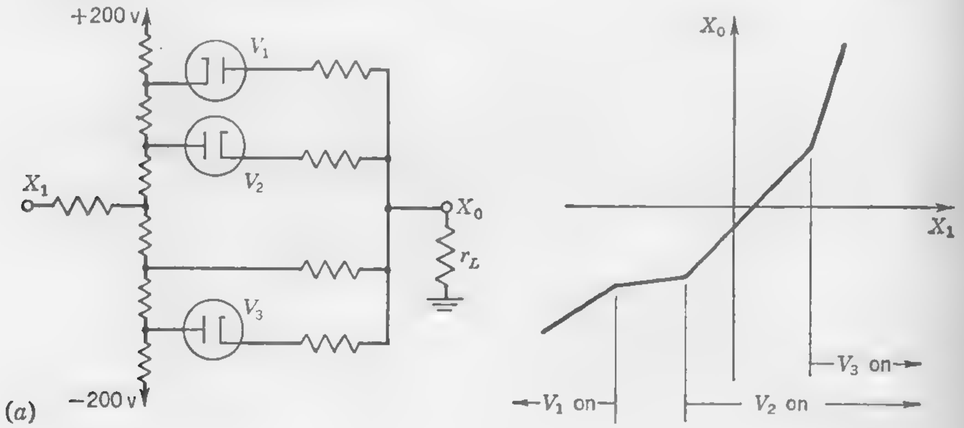


FIG. 3.3.11*a*, *b*. These maximum-remembering circuits combine amplitude comparison (Fig. 3.3.7) and analog storage (Sec. 4.2) to permit continuous read-out of max (X) as the input voltage X changes. Reversal of diodes and bias voltages would yield min (X). These circuits are also useful for accurate resetting of integrators in various timing or sweep circuits (see footnote to Art. 3.3.3).



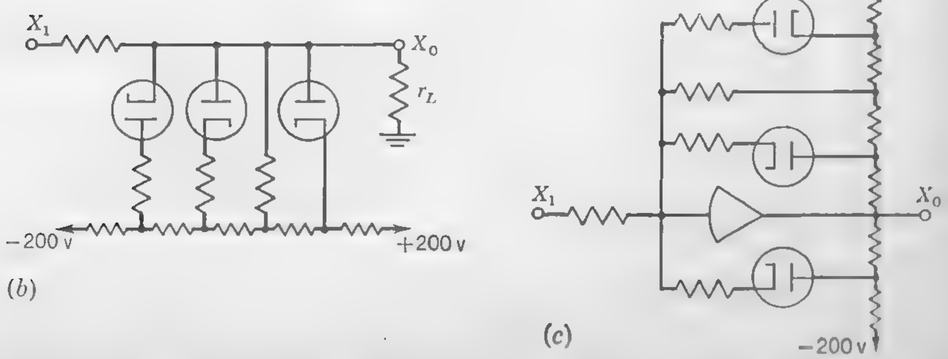


FIG. 3.3.12. Diode function generators for monotonic functions. Many combinations of such circuits are possible; nonmonotonic functions may be obtained as differences of monotonic functions.[2]

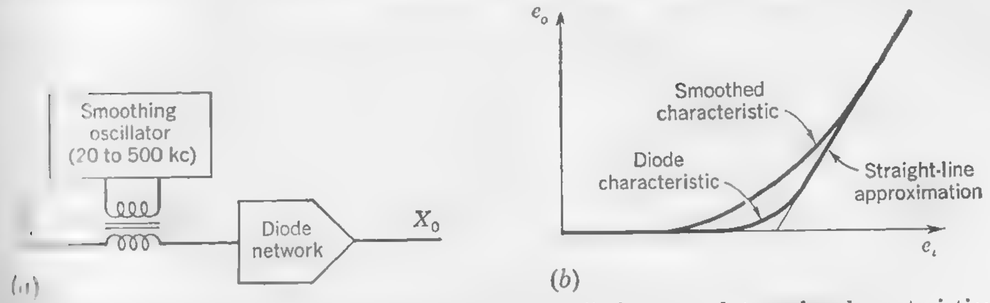


FIG. 3.3.13. Use of a smoothing oscillator to smooth diode-network transfer characteristics (*a*), and effect of smoothing on the voltage near a breakpoint (*b*) (based on Ref. 3).

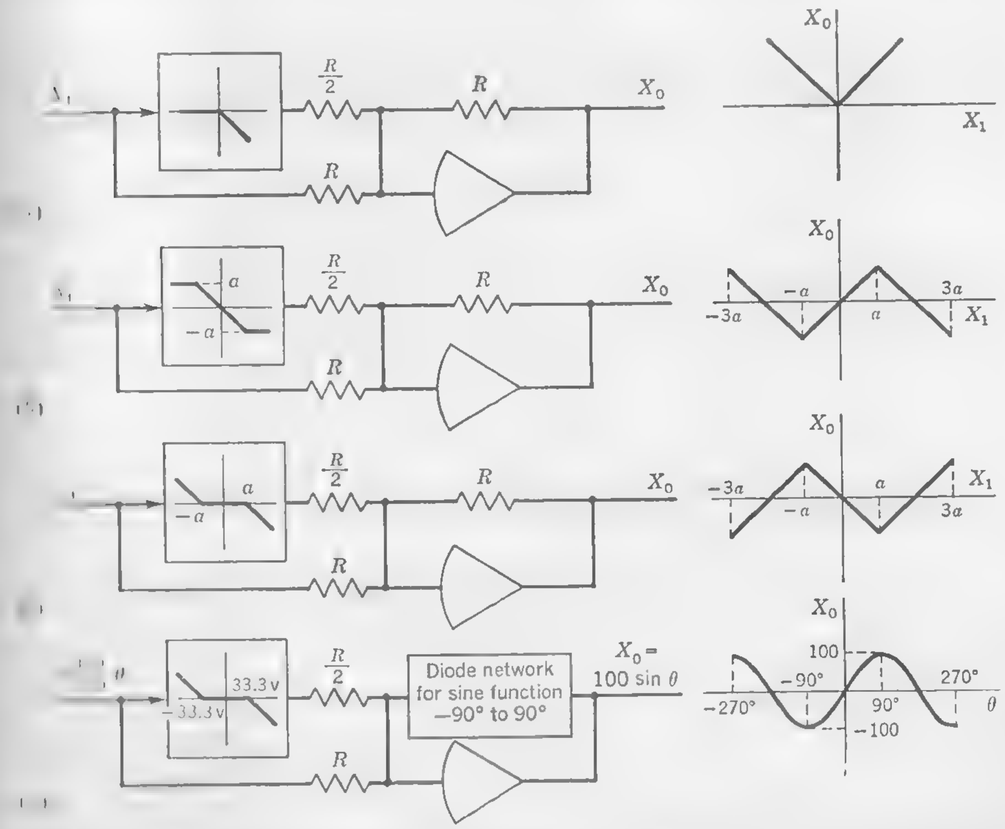


FIG. 3.3.14. Use of diode limiters to generate the absolute value (*a*) and quadrant-switching characteristics (*b*), (*c*) for sine-function generators (*d*). The precision-limiter circuits of Figs. 3.3.5 and 3.3.6 are especially useful in such applications.

coding the function generator only to correct an analytic approximation (Sec. 5.1), by adjusting *adjustable function masks* (Fig. 3.3.16*b*)[15] until the *measured* output voltage matches the desired function, or by preparing masks photographically with known deflection voltages.

MISCELLANEOUS FUNCTION GENERATORS

3.3.11. Varistor Function Generators.[16,17] The *varistor* function generator of Fig. 3.3.17 utilizes a current-sensitive nonlinear resistor (silicon-carbide varistor, e.g., General Electric *Thyrite* or Carborundum *Globar*) to produce a voltage transfer char-

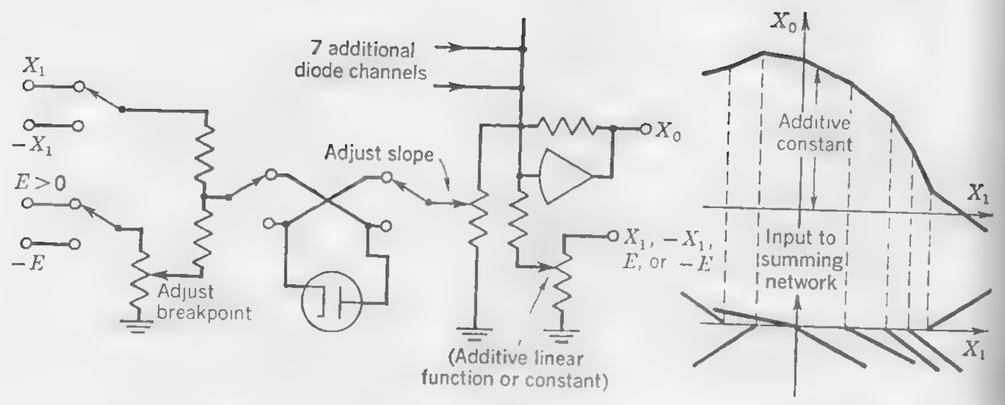


FIG. 3.3.15*a*. Simplified schematic diagram showing a series-limiter channel of the Reeves diode function generator. Different polarities of diode, input voltage, and bias voltage are chosen by a four-deck four-position "quadrant switch" yielding one of the four types of line segments shown at the right. Breakpoint and slope are set with a minimum of interaction on 10-turn helical potentiometer.

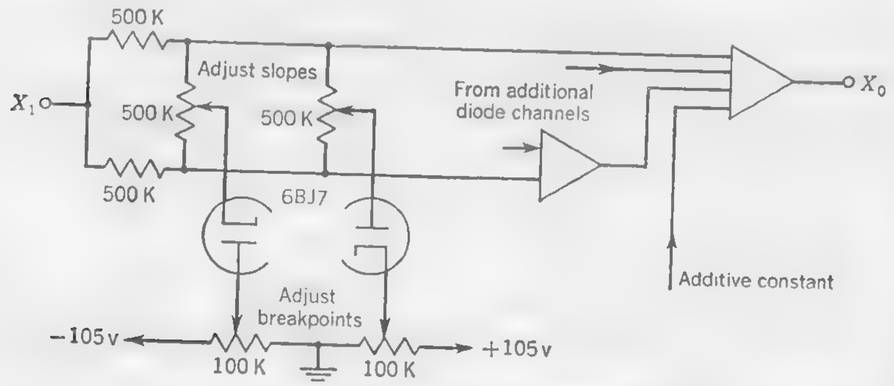


FIG. 3.3.15*b*. The Donner Model 35 diode function generator is an accessory package comprising 12 pairs of shunt limiters with regulated bias and filament supplies; two external d-c amplifiers are used.

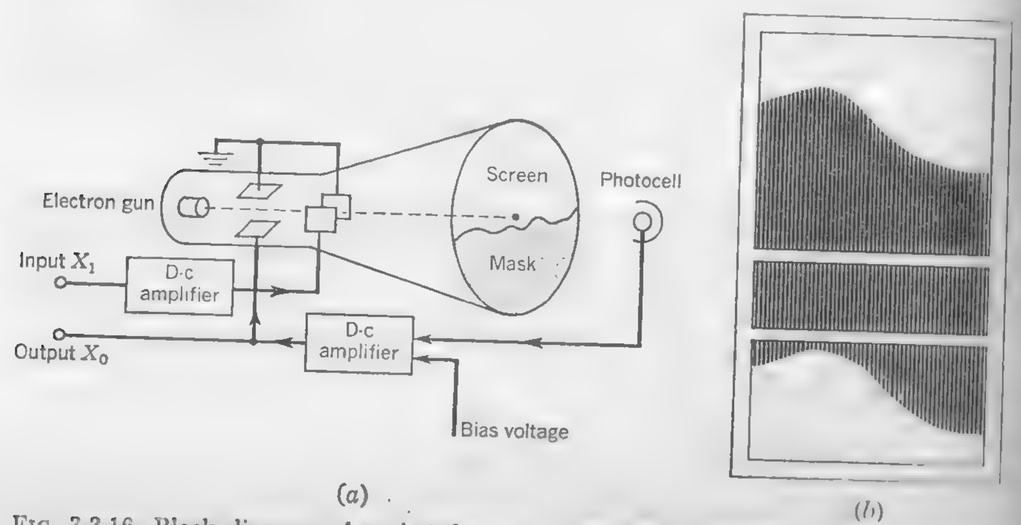


FIG. 3.3.16. Block diagram of a photoformer (*a*), and an adjustable photoformer mask (*b*) consisting of a set of metal shims clamped in a suitable frame (based on Ref. 15).

acteristic of the form $X_0 = aX_1^n$. In the Douglas Aircraft Company's Quadratron function units the factory-adjusted linear-resistance network sets the exponent n at 2 (P unit) or 3 (C unit), while a thermistor minimizes temperature-change effects on the varistor. The resulting transfer characteristic is accurate within 0.4 volt at d-c. Phase shift is comparable with that associated with an equivalent fixed-resistance network and is normally negligible below 100 cps for P units and below 50 cps for C units; with lead-network compensation quadratrons are usable up to 400 cps.

Varistor function generators are simple, rugged function generators of moderate cost and are useful within their accuracy limitations, particularly in special-purpose computers. Table 3.3.1 shows typical applications; combinations of P and C units produce *truncated power series* for many additional analytic functions.

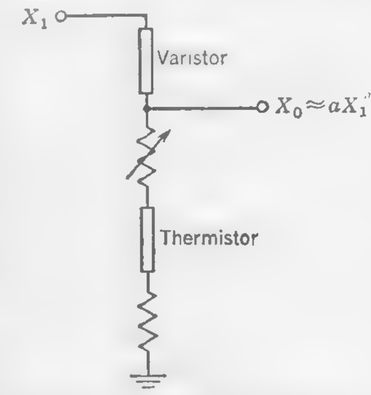


FIG. 3.3.17. Simplified circuit of a varistor function generator (Douglas Aircraft Co. QUADRATRON). Maximum ratings are 100 volts and 5 in A; maximum input resistance is 20 kilohms, maximum output impedance is 4 kilohms. RMS noise is of the order of 0.05 volt (based on Ref. 17).

3.3.12. Analog-digital Function Generators (see also Secs. 6.3, 6.5, and 7.1).[18] Various combinations of analog and digital computing techniques yield function generators generally characterized by high accuracy (0.1 to 0.01 volt) and convenience in function setup and storage. Specially designed analog-digital function generators can have excellent frequency response (to at least 1 kc), but then tend to become expensive. Methods of analog-digital function generation include the following:

1. Electronic switching of operational-amplifier gain-setting resistances by a voltage-controlled servotable binary counter ("digital servo"; see Sec. 7.1 for a multiplier circuit of this type). One counter can control several function generators.
2. Use of input and output analog-digital converters with (*a*) digital computation of analytic functions and/or (*b*) table look-up in digitally stored tables and digital interpolation. Either general-purpose or special-purpose digital computers can be used; refer to Sec. 6.5 for sampling and decoding problems.

Digital function generation is particularly useful in connection with functions of two or more variables (Art. 3.3.17). One can often reduce the effective cost of a digital function generator by time-sharing the digital function generator, i.e., by using one function generator to generate several functions (Sec. 6.3).

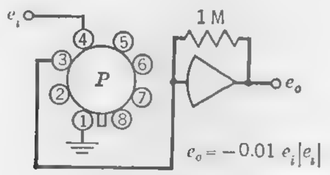
3.3.13. Low-frequency Test Oscillators. Low-frequency oscillators capable of generating sinusoidal (and sometimes also square-wave and triangular) signals over the frequency range of 0.01 to 2,000 cps are used to supply test excitation for various real or simulated control systems and for miscellaneous test applications. Desirable specifications include peak-to-peak output voltages of 40 to 50 volts, frequency calibration and sine-wave distortion within 2 per cent (preferably within 0.5 per cent), and some provision for phase-shifting or for phase-reference pulse output to facilitate phase measurements. Since LC and beat-frequency oscillators are not generally useful below 10 cps, the following types of oscillators are of interest:

1. Rate-servo-driven rotatable transformers (synchros) excited by 400 to 5,000 cps a-c, with or without phase-sensitive detection[2]
2. RC phase-shift oscillators
3. "Function-generator" types, which use a sine-generating diode network to shape the triangular waveform obtained through integration of a multivibrator square wave (Hewlett-Packard function generator)[19]

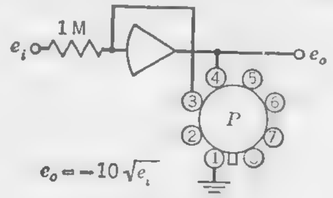
Rotatable-transformer oscillators[19] permit convenient phase shifting for servo testing, the modulated output waveform is also often useful in this connection. Such electromechanical devices are rarely used with d-c analog computers.

RC and function-generator-type test oscillators are of special interest for computer

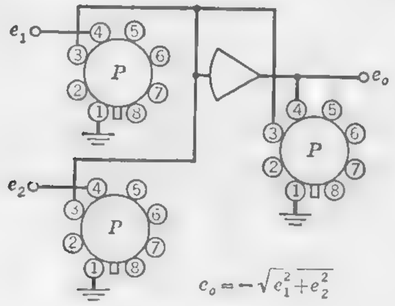
Table 3.3.1. Examples of Functions Generated with Varistor Function Generators*



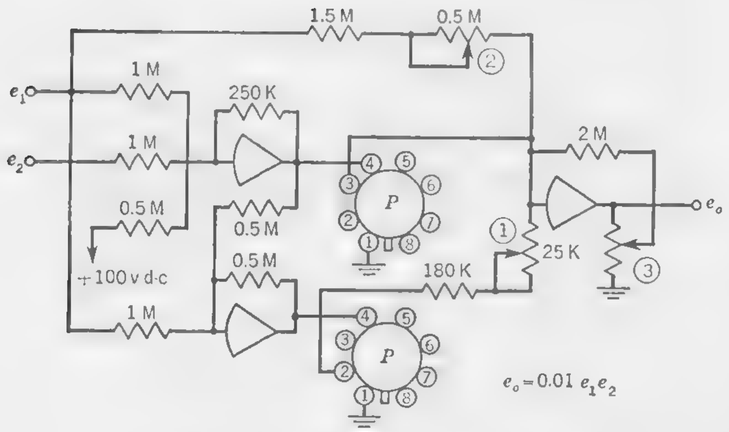
1. Square



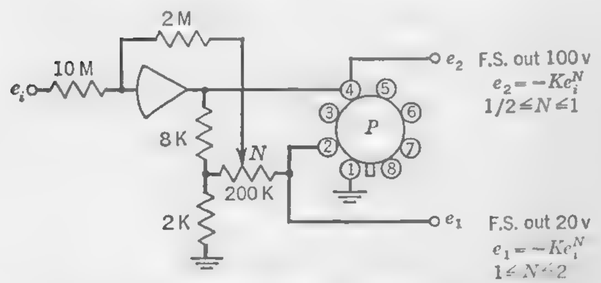
2. Square root



Right-triangle solver



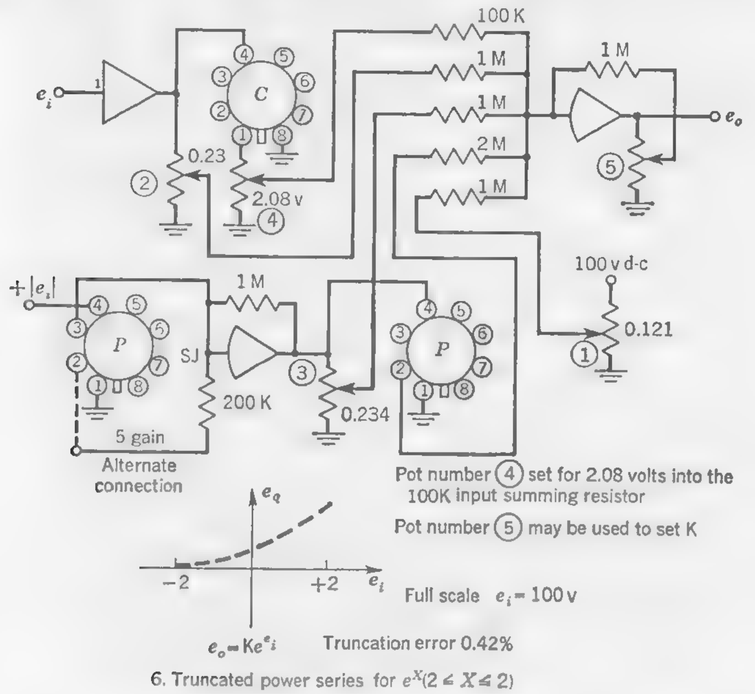
4. Quarter-square multiplier (can be modified for division; see also Sec. 3.2)



5. Powers with various exponents

*Contributed by L. D. Kovach and W. Comley, Douglas Aircraft Co., Santa Monica, Calif.; see Ref. 17 for additional examples, including truncated power series for trigonometric and hyperbolic functions.

Table. 3.3.1. Examples of Functions Generated with Varistor Function Generators. (*Continued*)



6. Truncated power series for $e^X (2 \le X \le 2)$

applications, because both can be implemented with standard operational amplifiers either on a computer patchbay or as special-purpose devices; the resulting accuracy is often *better* than that of commercially available test oscillators.

Figure 3.3.18 shows three examples of phase-shift oscillators. In practice, it is impossible to maintain the desired 90- or 180-deg phase shift at the signal frequency exactly, so that all phase-shift oscillators exhibit an increasing or decreasing amplitude (negative or positive damping). With high-quality amplifiers, and with variable resistors R replacing the phase-shift-generating potentiometers, the circuit of Fig. 3.3.18*a* can maintain its amplitude within 0.07 per cent per cycle at frequencies up to 70 cps. Practical circuits introduce automatic damping by a tungsten-filament lamp (Fig. 3.3.18*b*) or a thermistor, or a diode limiter circuit can be used to limit the amplitude. The distortion due to clipping is small (below 2 per cent) *if* the damping is critically adjusted, but this adjustment is not easy to maintain. Tungsten-filament damping introduces distortion below 20 cps because of insufficient thermal inertia; and indirect heating schemes with longer time constants tend to become complicated.

The Hewlett-Packard function generator uses a simple diode function generator to shape a multivibrator-generated triangular waveform into a sine-wave output voltage accurate to about 1 per cent. The free-running multivibrator/integrator is readily implemented with standard operational amplifiers (Fig. 3.3.9), so that addition of a simple −90- to +90-deg sine-generating network (Fig. 3.3.14) yields a very useful test oscillator setup capable of producing push-pull square-wave output and triangular waveforms as well as sine-wave output. In addition, differentiation of the square wave produces useful phase-reference markers. The timing accuracy is excellent, and the triangular output is invaluable for measuring transfer characteristics of function generators, transistors, etc., at different frequencies, and for testing multipliers (Sec. 3.7).

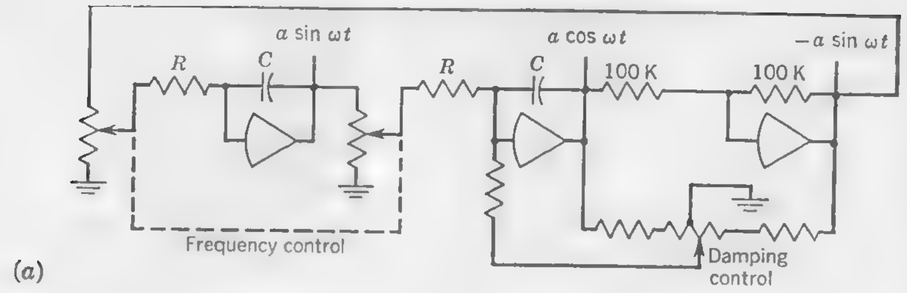


FIG. 3.3.18. Low-frequency test oscillators (phase-shift oscillators).

FUNCTIONS OF TWO OR MORE VARIABLES

3.3.14. Use of Conventional Computing Elements. Interpolation Schemes. A function of two or more machine variables X, Y, . . . is often obtained or approximated by means of judicious combination of summers, multipliers, and/or devices generating functions of a single variable. Thus one can obtain the function

$$F = F(X,Y) = \sin (X + Y)$$

by first adding the voltages X and Y and then generating the sine of their sum. Functions of several variables given by empirical data (tables or curves) can be similarly approximated by various approximation and interpolation schemes.[20 to 22] In particular, Levitt[22] suggests approximations of the form

$$F(X,Y) = f_1(X) + (Y - Y_1)f_2(X) + (Y - Y_1)(Y - Y_2)f_3(X) + \ldots \quad (3.3.1)$$

Three terms are often sufficient. The functions $f_1(X)$, $f_2(X)$, . . . are computed by successive substitution of $Y = Y_1, Y_2, \ldots$ in Eq. (1) and are set up on tapped potentiometers (Sec. 3.1) so that each product term can be generated by servo multiplication.

Figures 3.3.19 and 3.3.20 show two *mechanical interpolation schemes* adaptable for use with conventional analog computers.

Note that combinations of two-variable function generators can generate functions of any finite number of variables; thus $F(X,Y,Z) = F[f(X,Y),Z]$.

3.3.15. Use of Diode Function Generators.[23 to 25] If a variable voltage Y replaces the constant reference voltage E in the diode function generator of Fig. 3.3.15*a*, the output voltage X_0 will be a polygonal function of X for each value of Y (Fig. 3.3.21*a*), so that a function $F(X,Y)$ is produced.[23]

Since the break-point voltage of the ith limiter channel will be b_iY, where b_i is a constant, all break points generated by any one limiter channel must lie on a straight line through the origin. These *break-point loci* bound segments of equal slope and are indicated in broken lines in Fig. 3.3.21*a*.

One may generalize the above procedure by adding a function of X or Y to the output voltage X_0. One can obtain an even more general class of functions $F(X,Y)$ by introducing suitable functions $b_iG(Y)$ or $G_i(Y)$ of Y as bias voltages in the limiter channels of Fig. 3.3.15*a*, so that the break-point loci are no longer straight lines. To set up a given function $F(X,Y)$, one must locate the break-point loci by joining points of equal slope on a set of parametric lines like those in Fig. 3.3.21*a*.

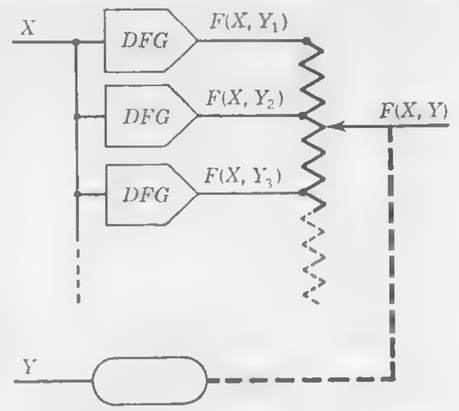


FIG. 3.3.19. Use of a servo-driven tapped potentiometer for interpolation.

The Philbrick Model F2V function generator (G. A. Philbrick Researches, Inc., Boston, Mass.)[24] uses a set of selector channels (Fig. 3.3.21*b*) to produce a desired function $F(X,Y)$ as a weighted sum of functions

$$F_i(X,Y) = \max [\min (X - X_i,\ Y - Y_i),\ 0]$$

which may be regarded as two-dimensional generalizations of the line segments in ordinary diode function generators. The contour lines of Fig. 3.3.21*c* illustrate the nature of a typical channel output $F_i(X,Y)$.

To set up a desired function $F(X,Y)$, one first chooses the break points (X_i,Y_i) and adjusts the slope controls to obtain the desired function values $F(X_i,Y_i)$ for successively increasing values of X_i and Y_i. Figure 3.3.21*d* shows a three-dimensional graph of a function $F(X,Y)$ generated by a Philbrick-type function generator.

The accuracy of the diode function-generator circuits described depends, again, on the nature of the function generated. Accuracies better than 1 per cent at frequencies in excess of 200 cps have been obtained in representative applications. The basic circuits shown permit many variations.

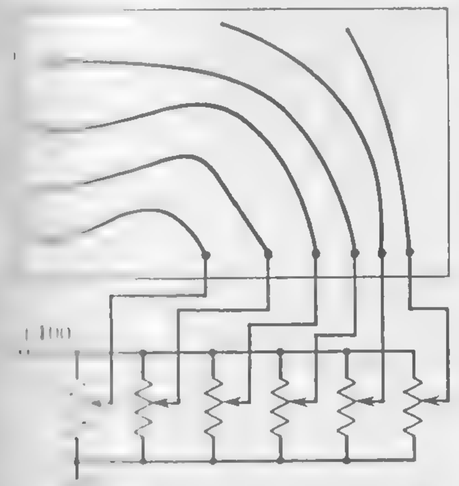


FIG. 3.3.20. Use of a servo plotting table with semiconducting paper for interpolation between contour lines $F(X,Y)$ = const drawn with conductive ink. A commercially available servo plotting table positions a probe sensing $F(X,Y)$; frequency response is limited to below 1 cps for ½-in. excursions. Function sheets can be stored for reuse.

3.3.16. Cathode-ray Function Generators. If a translucent mask of variable density is placed between the screen of a cathode-ray tube and a photocell[26] the photocell output voltage will vary as a function $F(X,Y)$ of the horizontal and vertical deflection voltages X and Y applied to the cathode-ray tube. The accuracy of such function generators is limited to between 1 and 10 per cent by the difficulty of positioning the beam accurately. The frequency response is excellent

(up to 100 kc), and the device is suitable for use in repetitive computers. Mask opacity variations over a range of 200 to 1 can be produced by photographic means.

A related more sophisticated type of cathode-ray function generator[2] employs a dual-beam cathode-ray tube. While the first beam travels over the function mask, a feedback circuit continuously positions the second beam on a calibrated comparison mask so that the light outputs from the two beams match; the deflection voltage of the comparison beam is the output voltage. A second feedback loop regulates the beam intensities.

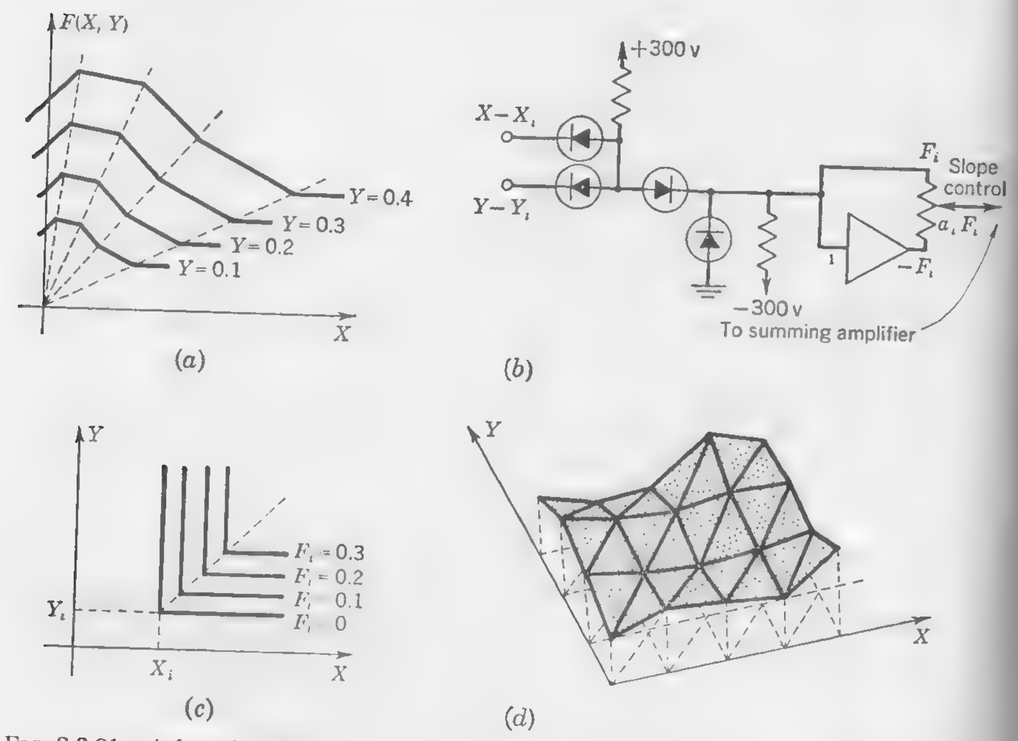


FIG. 3.3.21*a*. A function $F(X,Y)$ of two machine variables X and Y generated by the diode function generator of Fig. 3.3.15*a* if the constant reference voltage E is replaced by the variable voltage Y.

FIG. 3.3.21*b*. Generation of a function $F_i(X,Y) = \max [\min(X - X_i,\ Y - Y_i),0]$ by a double selection circuit. A very general class of functions may be approximated by sums $\sum_i a_i F_i$.

FIG. 3.3.21*c*. Contour lines of a function $F_i(X,Y) = \max [\min(X - X_i,\ Y - Y_i),0]$ generated by the circuit of Fig. 3.3.21*b*.

FIG. 3.3.21*d*. Approximation of a function $F(X,Y)$ by a sum of functions F_i of the type shown in Fig. 3.3.21*a*. Note the triangular facets.

Polimerou[27] has adapted his pulse-width-modulating cathode-ray function generator (Art. 3.3.9) to generate functions of two variables by a switching scheme selecting one of several transparent function lines in accordance with a second voltage input.

3.3.17. Analog-digital Function Generators. Analog-digital function generators can produce functions of two variables by digital operations and/or table look-up and interpolation essentially as easily as functions of a single variable (Art. 3.3.12, Secs. 6.3 and 6.5). This application is, indeed, often the main reason for adopting analog-digital function generators; digital function setting and storage are particularly convenient.

RANDOM-NOISE GENERATORS

3.3.18. Noise-generator Specifications.[28,29] Analog-computer studies of random processes and related statistical studies require sources of randomly varying voltages (noise) for use as random driving functions, coefficients, and/or initial conditions (Secs. 5.9 and 6.3). Wide-band noise of approximately gaussian character is easily obtained through amplification of the output of noisy resistors (thermal noise), temperature-limited vacuum diodes (shot effect), gas-filled tubes, and of Geiger-Mueller or scintillation counters with a small radioactive source. It is quite another matter to produce noise which satisfies the accurate statistical specifications desirable in typical computer applications, viz.,

Gaussian or other specified probability distribution within 1 to 2 per cent probability

8-hr average d-c unbalance at output less than 0.05 volt with probability 0.95 ("slow" computers)

Maximum rms output level of the order of 10 volts, with 0.2 per cent calibration accuracy

Power spectrum flat within 0.1 db to 35 cps ("slow" computers) or within 2 db to 25 kc (repetitive computers, Sec. 6.3)

Output uncorrelated between samples: autocorrelation function (Sec. 5.9) less than 1 per cent of mean-square output for delays exceeding 0.02 sec ("slow" computers) or 0.05 msec (repetitive computers)

In particular, it is necessary to discriminate carefully against the effects of line-frequency pickup and power-supply variations which might introduce periodic components into the noise-generator output, since the effects of such errors can accumulate in statistical averaging operations. For the same reason, even the measurements required to check the above statistical specifications are subject to pitfalls. Since very long integration times would be required for accurate checks of low-frequency noise generators it is probably best to sample the noise-generator output accurately and to compute the requisite statistics digitally from the voltage samples.

White-gaussian-noise generators can be made to yield noise with various power spectra and probability distributions by the methods of Sec. 5.9.

3.3.19. Noise Generators Employing Radioactive Sources. In Fig. 3.3.22*a* a self-quenched Geiger-Mueller counter tube activated by a small radioactive source triggers a bistable multivibrator at each count to produce a random telegraph wave, with the probability of K zero crossings during any time interval T given by the Poisson distribution

$$p(K,T) = \frac{(\alpha T)^K}{K!} e^{-\alpha T}$$

where α is the mean count rate. The random telegraph wave can be accurately limited and may be passed through a low-pass filter to yield band-limited white gaussian noise. Figure 3.3.22*b* shows a practical circuit. The random square-wave output is useful in its own right, particularly in simulations involving randomly timed events (processes with equipment failures, waiting-line problems).[33,34]

The useful frequency range of such noise generators is limited by the mean count rate α, and thus by the deionization time of the Geiger-Mueller tube; in particular, the cutoff frequency of the low-pass filter in Fig. 3.3.22*a* must be small compared with α if a flat output spectrum is desired. Substitution of a photomultiplier-type *scintillation counter* and pulse amplifier for the Geiger-Mueller tube can increase the useful frequency range up to 30 kc, so that the noise generator can be used with repetitive analog computers. The long-term average count rate can be stabilized through integral-control feedback of rectified pulses to the photomultiplier power supply and/or the multivibrator trigger level, but the sensitivity of photomultipliers to power-supply variations and stray magnetic field presents problems.[35]

3.3.20. Demodulator- and Sampling-type Noise Generators.[28 to 32] The *demodulator type noise generator* shown in Fig. 3.3.23 was originally developed at the Hughes Aircraft Company and is packaged for three major analog-computer manufacturers by

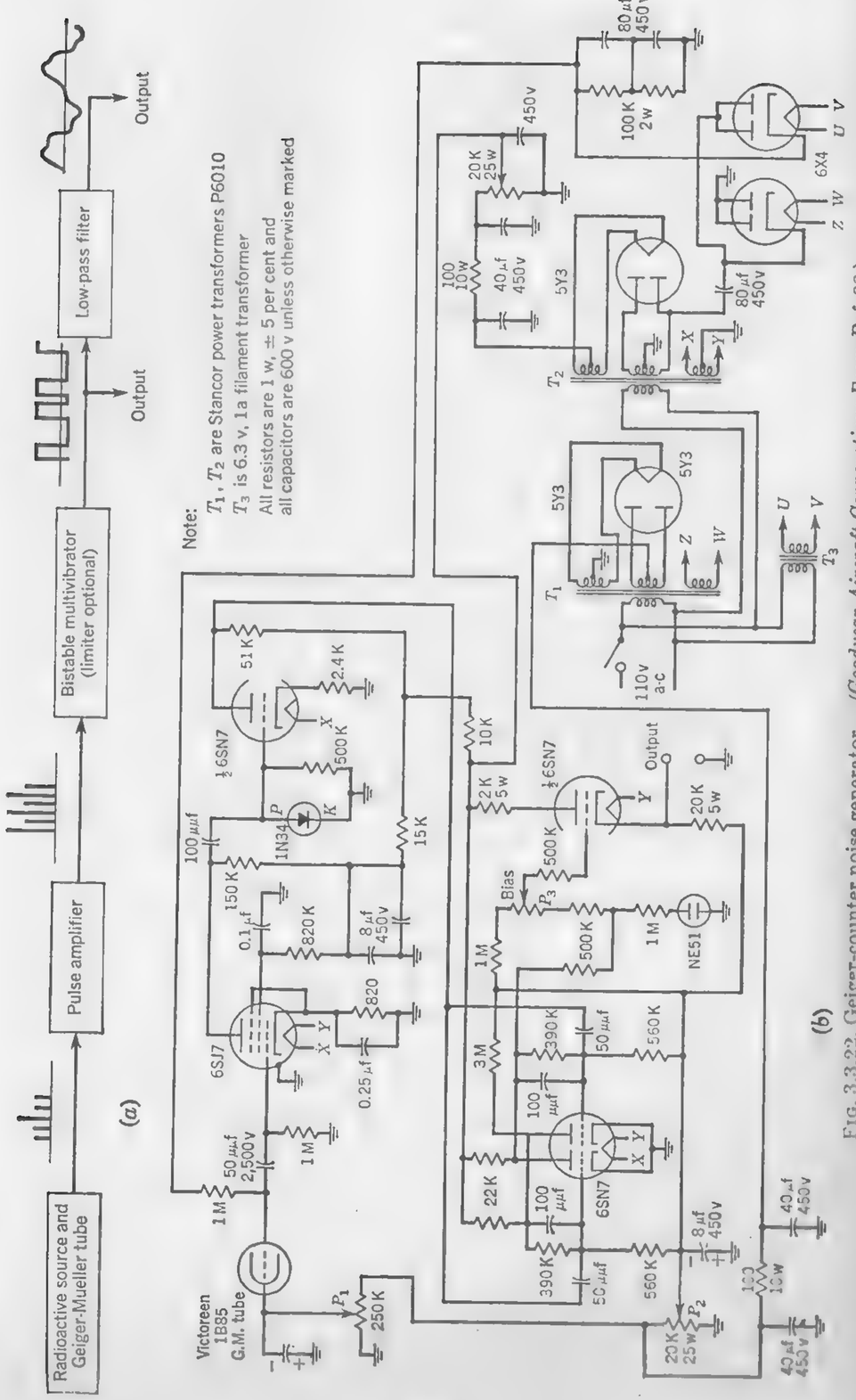


Fig. 3.3.22. Geiger-counter noise generator. (*Goodyear Aircraft Corporation; From Ref. 33.*)

Elgenco, Inc. (Los Angeles, Calif.). Noise from a thyratron placed in a transverse magnetic field to prevent resonance peaks is first amplified by an a-c regulator amplifier whose gain is accurately controlled to yield a constant rms output. A bandpass amplifier selects a band of frequencies about 100 cps wide and accurately centered about 400 cps. The resulting regulated, band-limited noise is demodulated by a 400-cps chopper (synchronous demodulator) and filtered to produce an output spectrum flat within 1 db to 35 cps and centered around zero frequency. Note that the demodulation process eliminates any low-frequency fluctuations introduced in the noise source and low-level amplifier stages. The Elgenco noise generator satisfies the specifications

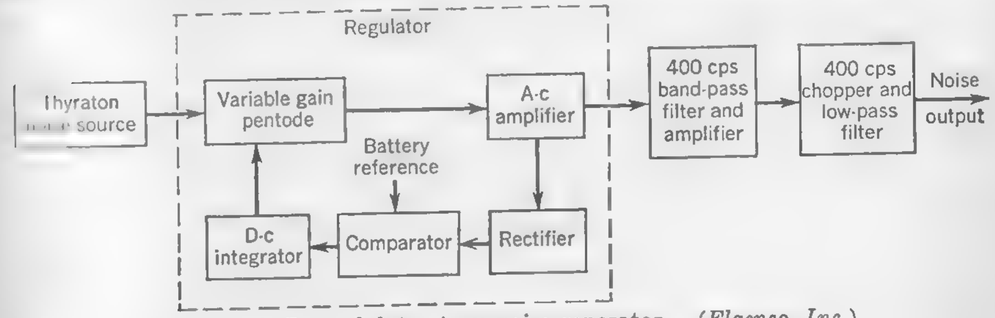


Fig. 3.3.23. Demodulator-type noise generator. (*Elgenco, Inc.*)

for low-frequency noise generators listed in Art. 3.3.18; it is not suitable for repetitive-computer applications.

Sampling-type noise generators[31,32] replace the demodulator block in Fig. 3.3.23 by a sample-hold circuit (Sec. 6.3) and low-pass filter. The sampling frequency may be of the order of 400 cps and must not be commensurable with the line frequency.

References

Diode Circuits

1. Morrill, C. D., and R. V. Baum: Diode Limiters Simulate Mechanical Phenomena, *Electronics*, November, 1952.
4. Korn, G. A., and T. M. Korn: *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.
4. Smith, G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1959.
4. Medkeff, R. J., and R. J. Parent: A Diode Bridge Limiter for Use with Electronic Analog Computers, *Proc. AIEE*, **70**: Sec. T-1-170 (1951).
5. Koerner, H., and G. A. Korn: Function Generation with Operational Amplifiers, *Electronics*, Nov. 6, 1959.
6. Korn, G. A.: A Precision Dead-space Limiter for Triangle-integration Multipliers, Electrical Engineering Department, University of Arizona, *ACL Mem.* 14, November, 1959.
8. Howe, R. M.: Representation of Nonlinear Functions by Means of Operational Amplifiers, *IRE Trans. PGEC*, December, 1956.
9. Tomlinson, N. P.: Fundamental Circuits and Techniques Used with Electronic Analog Computers, Goodyear Aircraft Company, Akron, Ohio, *Rept.* AP-77079, 1956.
9. Diamantides, N. D.: A Multipurpose Electronic Switch for Analog Computer Simulation and Autocorrelation Applications, *IRE Trans. PGEC*, December, 1956.

Cathode-ray Function Generators

[illegible], D. E.: Photoelectric Waveform Generator, *Electronics*, **22** (February, 1949).
[illegible], E. J.: Photoformer Design and Performance, *Proc. Natl. Electronics Conf.*, 7:188 (1951).
10. Gerlach, A. A., C. N. Pederson, and R. E. Zenner: A Precise Electronic Function Generator, *Proc. Natl. Electronics Conf.*, **8**.
11. Palimeron, L. G.: A New Method of Generating Functions, *IRE Trans. PGEC*, September, 1954.

14. Silverberg, B.: Function Generator for Radar Simulator, *Electronics*, Jan. 9, 1959.
15. MacNee, A. B.: A High-speed Product Integrator, *MIT Research Lab. Electronics Rept.* 136, Aug. 17, 1949.

Miscellaneous Function Generators

16. Kovach, L. D., and W. Comley: An Analog Multiplyer Using Thyrite, *IRE Trans. PGEC*, June, 1954.
17. Kovach, L. D., and W. Comley: Nonlinear Transfer Functions Using Thyrite, *IRE Trans. PGEC*, June, 1958.
18. Grabbe, E. M., S. Ramo, and D. E. Wooldridge: *Handbook of Automation, Computation, and Control*, vol. 2, John Wiley & Sons, Inc., New York, 1959.
19. Cockrell, W. D.: *Industrial Electronics Handbook*, sec. 6, McGraw-Hill Book Company, Inc., New York, 1958.

Functions of Two or More Variables

20. Jerrard, R. P., and G. T. Jacoby: Generation of a Function of Two Variables, ACM Conference Paper, Cambridge, Mass., September, 1953.
21. Kindle, W. K.: Method of Approximating Functions of Two Variables with Functions of Single Variables, Electronic Associates, Inc., Long Branch, N.J., *PCC Rept.* 53, July 2, 1956.
22. Levitt, J. R.: Bi-variable Function Generation, AIEE Conference paper CP 57-311, Jan. 21, 1957.
23. Meissinger, H. F.: An Electronic Circuit for the Generation of Functions of Several Variables, *IRE Natl. Convention Record*, 1955.
24. Data Sheet for Model F2V Function of Two Variables Component, G. A. Philbrick Researches, Inc., Boston, Mass., November, 1957.
25. Stern, T. E.: *Piecewise-linear Network Theory, MIT Research Lab. Electronics, Rept.* 315, June 15, 1956.
26. Wallmann, H.: An Electronic Integral-transform Computer and the Practical Solution of Integral Equations, *J. Franklin Inst.*, **250**:45 (1950). Variable Density Film Yields Three Functions, *Control Eng.*, May, 1956.
27. Polimerou, L. G.: A New Method for Generating a Function of Two Independent Variables, *IRE Trans. PGEC*, September, 1957.

Random-noise Generators

28. Bennett, R. R., and A. S. Fulton: The Generation and Measurement of Low-frequency Random Noise, *J. Appl. Phys.*, **22**:1187 (September, 1951).
29. Beecher, D. E., R. R. Bennett, and H. Low: Stabilized Noise Source for Air-weapons Design, *Electronics*, July, 1954.
30. Favreau, R. R.: Noise Generators, Electronic Associates, Inc., Long Branch, N.J., *Princeton Computation Center Rept.* 19.
31. Karpeles, M. A.: Noise Compressor, U.S. Patent No. 2,649,386, dated May 19, 1953.
32. Winter, D.F.: A Gaussian Noise Generator for Frequencies Down to 0.001 cps, *IRE Natl. Convention Record*, **23**: part 4 (1954).
33. Random Noise Generator for Simulation Studies, Goodyear Aircraft Corp., Akron, Ohio, *Rept.* GER-6436, Dec. 13, 1954.
34. Two Applications of GEDA computers to Statistical Problems of Operations Research, Goodyear Aircraft Corp., Akron, Ohio, *Rept.* GER-6729, June 4, 1955.
35. Hartmann, J., and H. Koerner: Electrical Engineering Department, University of Arizona, unpublished memorandum, 1960.

Section 4

ELECTRONIC ANALOG COMPUTERS: CONTROL CIRCUITS, COMPUTER OPERATION, AND SYSTEM DESIGN

By

BERNARD D. LOVEMAN, *Project Engineer, Project CYCLONE, Reeves Instrument Corporation, Garden City, N.Y.*

THADDEUS J. KUSTO, *Director of Research, Computer Systems, Inc., New York, N.Y.*

GRANINO A. KORN, *Professor of Electrical Engineering, University of Arizona, Tucson, Ariz.*

STANLEY ROGERS, *Head, Data Systems, General Dynamics/San Diego, San Diego, Calif.*

HAROLD L. EHLERS, *Supervisor, Flight Control Section, Autonetics Division, North American Aviation, Inc., Downey, Calif.*

WALTER HOCHWALD, *Supervisor, Inertial Navigation Department, Autonetics Division, North American Aviation, Inc., Downey, Calif.*

CONTENTS

PART 1: CONTROL CIRCUITS

Bernard D. Loveman

Power Controls

4.1.1. Basic Power-control Circuits. 4–3
4.1.2. A More Refined Power-control System. 4–4

Operating Controls

4.1.3. Introduction. 4–4
4.1.4. RESET and OPERATE (COMPUTE) Modes. 4–4
4.1.5. HOLD Mode. 4–5
4.1.6. Integrator Relays. 4–7
4.1.7. BALANCE CHECK (POT SET) Mode. 4–8
4.1.8. REFERENCE OFF Mode. 4–9
4.1.9. REPETITIVE Mode. 4–9
4.1.10. SLO-RESET Mode. 4–9

Miscellaneous Control Functions

4.1.11. Control Panels. 4–9
4.1.12. Slave (Parallel) Operation. 4–11
4.1.13. Remote Control. 4–11
4.1.14. Individually Controlled Integrators. 4–11

PART 2: SEMIAUTOMATIC PROGRAMMING

Bernard D. Loveman

Introduction

4.2.1. Survey 4-12

Automatic Programming Equipment

4.2.2. Autocycler 4–12
4.2.3. Output Scanner 4–12
4.2.4. Input-output Programmer 4–13
4.2.5. Recorder Calibration and Controls 4–15

Automatic Programming Techniques Using Analog Storage

4.2.6. Analog-storage Circuits 4–17
4.2.7. Automatic Scale Changing 4–17

PART 3: PROBLEM-CHECKING CIRCUITS AND PROCEDURES

Bernard D. Loveman

Introduction

4.3.1. Survey 4–18

Excessive and Insufficient Signals

4.3.2. Amplifier Overload Circuits . . . 4–19
4.3.3. Computer-servo Overload Indicators 4–20
4.3.4. Minimum-excursion Indicators 4–20

Programming Checks

4.3.5. Patchboard Verifiers 4–21
4.3.6. Static Checks 4–23
4.3.7. Integrating-capacitor Tests . . . 4–24
4.3.8. Time-scale Checks 4–24

PART 4: PATCHBAYS, INTERCONNECTIONS, AND ENCLOSURES

Thaddeus J. Kusto

Patchbays

4.4.1. Introduction 4–26
4.4.2. Patchboard Layout 4–26
4.4.3. Cellular or Shielded Construction 4–26
4.4.4. Molded Construction 4–28
4.4.5. Patchboard Layout and Human Engineering 4–29

Interconnections and Ground Circuits

4.4.6. Miscellaneous Design Considerations 4–30
4.4.7. Leakage 4–30
4.4.8. Parasitic Coupling Impedances and Noise 4–31
4.4.9. Ground Circuits 4–31
4.4.10. Stray Capacitances 4–32

Computer Enclosures

4.4.11. Cabinets 4–33
4.4.12. Ventilation 4–33

PART 5: COMPLETE ELECTRONIC ANALOG COMPUTERS: SMALL AND MEDIUM-SIZED COMPUTERS

Granino A. Korn

Introduction

4.5.1. Introductory Remarks. Applications 4–34
4.5.2. Design for Expansion 4–34

System Design

4.5.3. Amplifiers and Power Supplies. 4–35
4.5.4. Nonlinear Computing Elements 4–35
4.5.5. Patchbays, Simulation Boards, and Control Systems 4–35
4.5.6. Control Systems for Small Computers 4–38
4.5.7. Metering System and Recorders 4–38
4.5.8. Reference Power Supplies 4–38
4.5.9. Typical Installations 4–38

PART 6: COMPLETE ELECTRONIC ANALOG COMPUTERS: LARGE INSTALLATIONS

Stanley Rogers

Introduction

4.6.1. Large Analog Computers 4–40

Major Design Objectives

4.6.2. Requirements 4–40
4.6.3. Simultaneous Solution of Different Problems 4–40
4.6.4. Efficient Use of Computing Elements 4–40
4.6.5. Accuracy Requirements for Large Problems 4–41

4.6.6. Reliability and Checking Features 4–41

Problem Setup and Storage

4.6.7. Requirements 4–41

Personnel and Administration. Typical Installations

4.6.8. Operators 4–41
4.6.9. Administration of Large Computer Facilities 4–42
4.6.10. Statistics on Selected Computer Installations 4–44

PART 7: MAINTENANCE PROCEDURES AND TEST CIRCUITS

Harold L. Ehlers and Walter Hochwald

Introduction

4.7.1. Planned Maintenance 4–44

Test Circuits

4.7.2. Operational Amplifiers 4–44
4.7.3. Reference-voltage Supplies . . . 4–46
4.7.4. Passive Elements and Control Relays 4–46
4.7.5. Servomultipliers 4–52
4.7.6. Servo Resolvers 4–52
4.7.7. Electronic Multipliers 4–52
4.7.8. Coefficient Potentiometers . . . 4–53
4.7.9. Diodes and Operational Relays 4–53

Test Equipment

4.7.10. General Considerations 4–54
4.7.11. Test Benches 4–55
4.7.12. Auxiliary Instrumentation . . . 4–55
4.7.13. Test Boards 4–56

Maintenance Procedures

4.7.14. Types of Maintenance Operations 4–56
4.7.15. Preventive Maintenance 4–56
4.7.16. Routine Maintenance 4–57
4.7.17. Emergency Maintenance 4–57

Maintenance Records

4.7.18. Function and Significance of Records 4–57
4.7.19. Examples 4–57
4.7.20. Equipment Identification 4–58

Spare Parts

4.7.21. Spare-parts Supply 4–58

PART 1: CONTROL CIRCUITS

By Bernard D. Loveman

POWER CONTROLS

4.1.1. Basic Power-control Circuits. Figures 4.1.1 and 4.1.2 show basic *power-control circuits*. In each case, filament supplies are energized before high-voltage (plate) supplies; chopper excitation may be applied together with (or just before) high

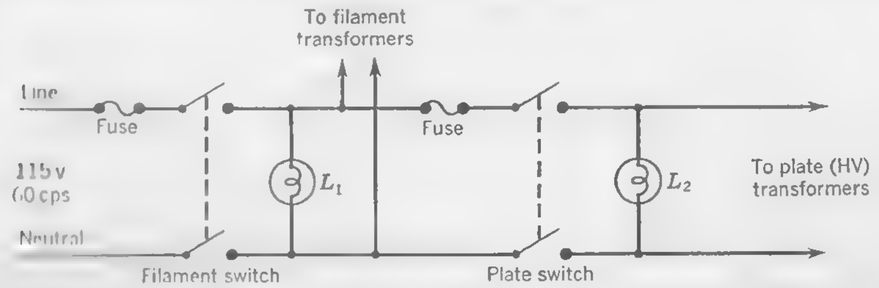


Fig. 4.1.1. Basic power-control circuit using toggle switches (up to 20 amp).

voltage to save contact wear. The circuit of Fig. 4.1.2 has push buttons and latching relays, so that one does not need to check the high-voltage switch before energizing the filaments.

4.1.2. A More Refined Power-control System. Another power-control circuit used on at least one large computer installation consists of a servo-driven autotransformer which increases the filament voltage gradually from zero to rated voltage to prevent vacuum-tube failures due to thermal shock. When the main switch is first turned on the voltage to the computer filament and plate transformers stays off until the servo has slewed the arm of the autotransformer to its zero position. Then the output of the autotransformer is connected to the computer filaments and the applied voltage is increased to 95 volts over a 3-min interval. Next, the servo is switched to its positioning mode and nulls at 115 volts ± 1 volt. Finally, a contactor closes to permit plate power to be applied.

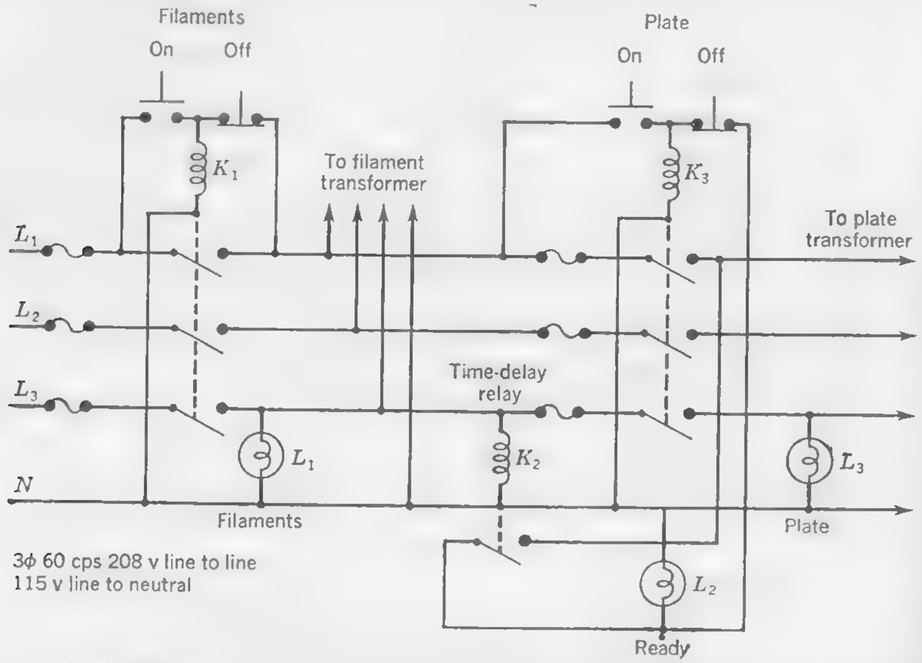


FIG. 4.1.2. Typical power-control circuit for large systems. Momentary ON-OFF push buttons are used. K_2 is a time-delay relay; the plate voltage cannot be turned ON until about 3 min after filament voltage is applied.

OPERATING CONTROLS

4.1.3. Introduction. *Computer operating modes* are required to control integrator operation; i.e., the initial conditions must be introduced, and the computation must be started and stopped. The two essential modes are RESET and OPERATE (COMPUTE); in the former the initial conditions are applied and in the latter integration takes place and the computer gives a solution to the problem. In addition several other modes are usually provided to permit efficient operation of the computer; these are HOLD, BALANCE CHECK (POT SET), and PROBLEM CHECK (see also Sec. 4.3). Finally, some analog computers have special-purpose modes such as REFERENCE OFF, REPETITIVE, and SLO-RESET.

4.1.4. RESET and OPERATE (COMPUTE) Modes. To set initial conditions, one can discharge each integrating capacitor to zero and add a d-c voltage equal to the desired initial condition to the integrator output (RESET). To start the computation, one releases the integrating capacitors, with the initial-condition voltage still added in (Fig. 4.1.3). Because of the complication of the extra addition, this method is used mainly on high-speed (repetitive) computers (Sec. 6.2).

In the second method, the inputs of each integrator are removed and each integrating condenser is charged, so that the integrator output voltage will start from the correct initial value (RESET). To start the computation, one removes the initial-condition voltages and reconnects the inputs (Fig. 4.1.4). In RESET, it is important to remember that dielectric absorption (soakage) will produce errors in the initial condition. For high-quality polystyrene condensers, this error can be held to 10 mv, provided that the integrator is in RESET for at least 30 sec (see also Sec. 2.4).

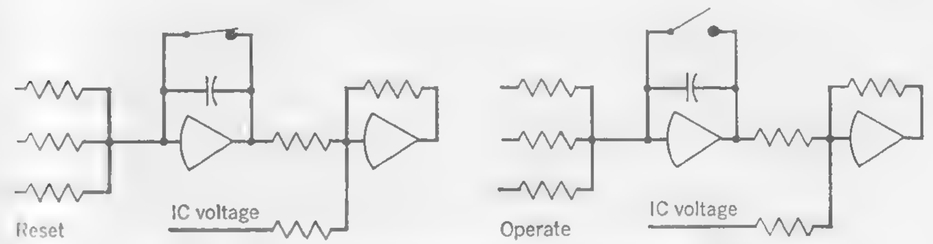


FIG. 4.1.3. A reset circuit used only for repetitive operation.

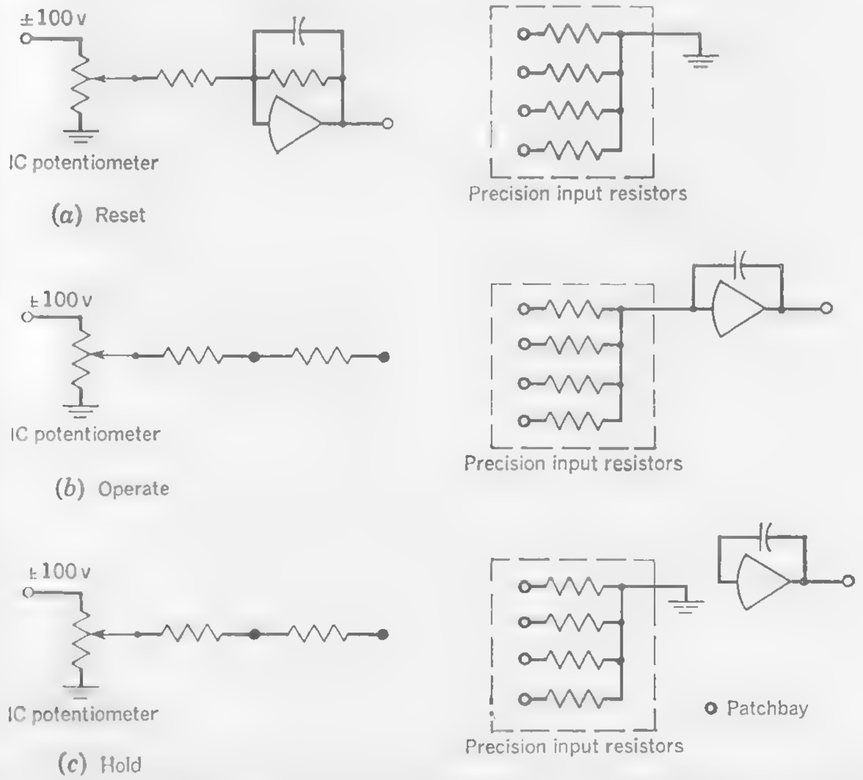


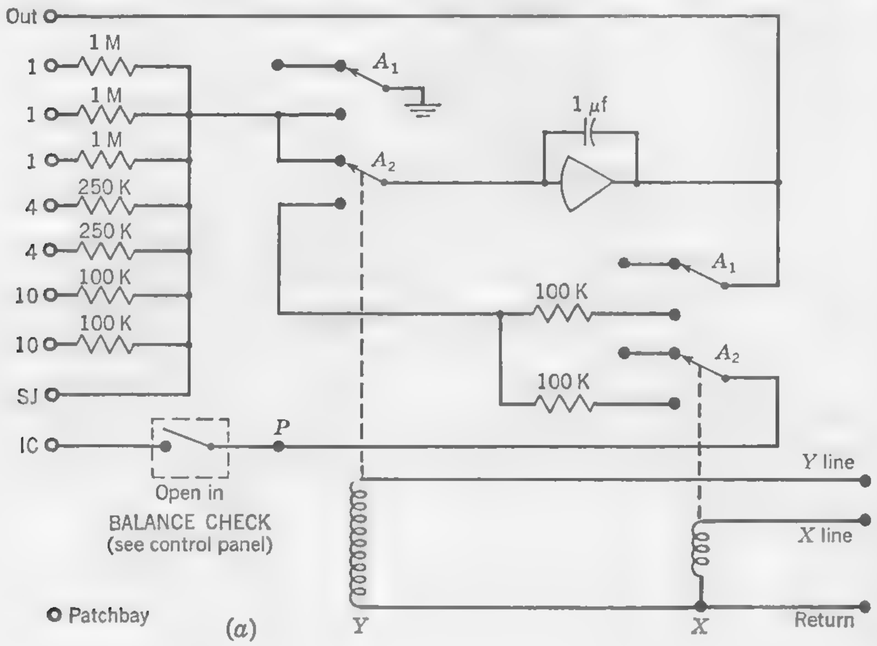
FIG. 4.1.4. Standard integrator modes (*a*) RESET, (*b*) OPERATE (COMPUTE), (*c*) HOLD.

The voltages at other components such as summing amplifiers, position servos (multipliers or resolvers), function generators, and electronic multipliers are determined at all times by the output of the integrators and/or by fixed constants. In RESET and OPERATE—as well as in HOLD—these components are in "operate." If the computer contains electromechanical integrators such as rate servos or time motors, their modes must be controlled. Each one has an initial position from which it starts when the computation begins, and because of inertia the initial rate should be zero (see also Sec. 3.1).

4.1.5. HOLD Mode. This mode is used to halt operation of the computer in such a fashion that all voltages remain at the value they had at the instant the computation was stopped. Using this mode, the operator can measure the instantaneous value of

one or more of the problem variables at a specified time, or at the time another variable exceeds or falls below a given voltage. This mode is also used to stop a problem so that the circuits or coefficients can be changed.

To halt an electronic integrator its inputs are disconnected from the amplifier; the output voltage will then remain fixed, provided the feedback condenser retains its



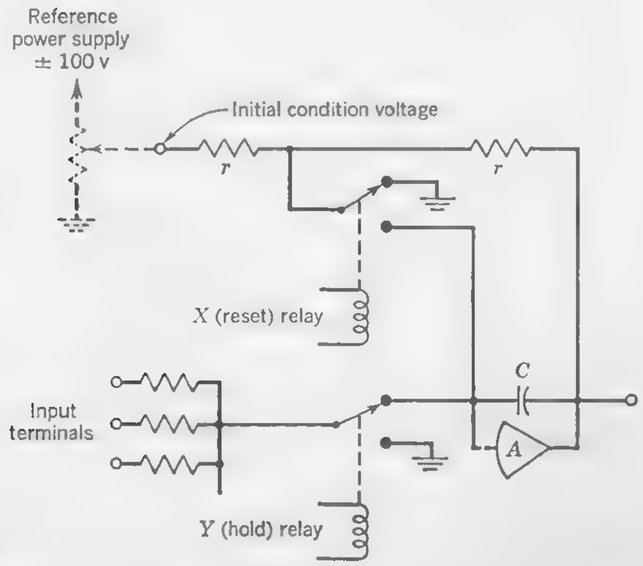


FIG. 4.1.5. Two integrator relay circuits. RESET: relays X and Y are energized. OPERATE: relays X and Y are deenergized. HOLD: relay X is deenergized and relay Y is energized.

charge (Fig. 4.1.4*c*). In this mode the condenser discharge (or charge) rate is determined by its leakage resistance and by the summing-junction current. The latter is due to grid current, amplifier unbalance, and rectified a-c noise from the chopper. With a quality polystyrene condenser and a high-performance d-c amplifier this drift can be made as small as 1 mv per min.

In HOLD the voltages of the other components such as summing amplifiers, function generators, and electronic multipliers are held constant by the integrator output voltages. Position and rate servos have inertia and overshoot when the computer is switched to HOLD. An error signal is then generated to permit the servo to recover. In position servos the error is developed because the input signal stops; in rate servos an integrating circuit is added in HOLD to produce the required error voltage.

4.1.6. Integrator Relays. Since the d-c analog computer is a parallel machine, all integrators must ordinarily start and stop at the same instant. "Slow" d-c

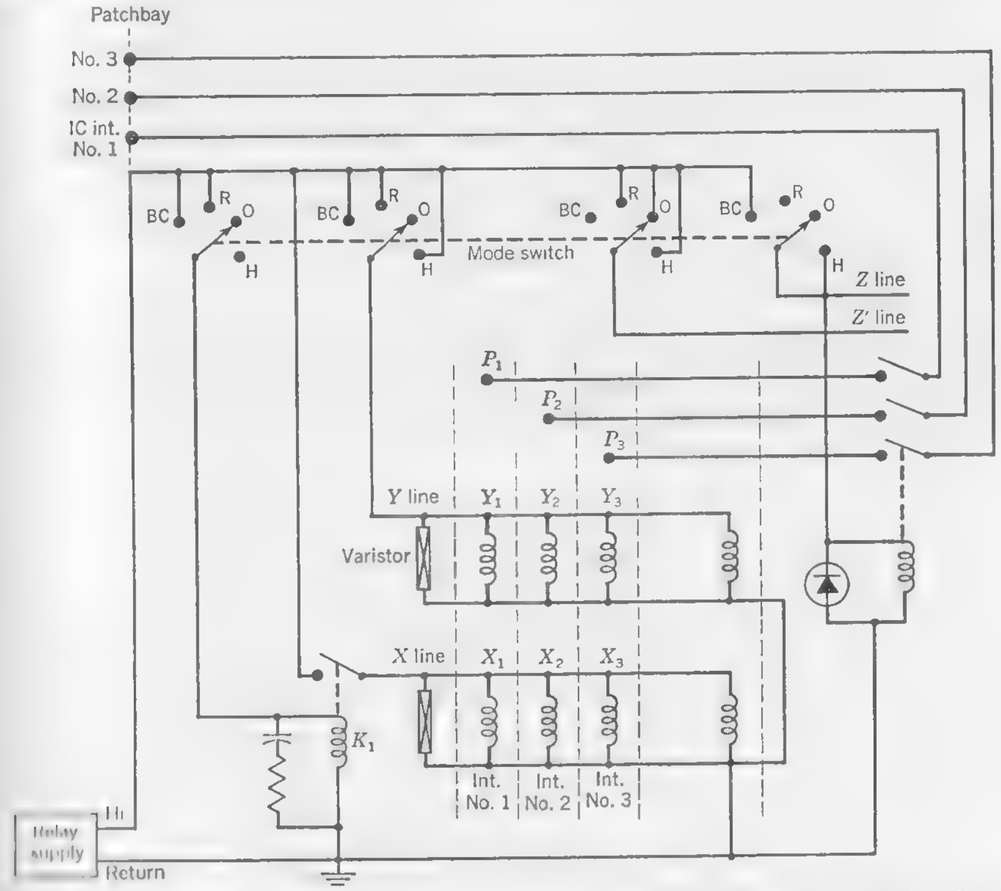


FIG. 4.1.6. Computer mode-control system. In BALANCE CHECK the X, Y, and Z lines are energized and initial-condition inputs are disconnected. In RESET X, Y, and Z' lines are energized. In OPERATE only Z' line is energized. In HOLD Y and Z' lines are energized. Varistors are used to suppress transients on the relay lines. The rotary mode switch shown can be replaced by suitable push buttons.

analog computers use relays to change integrator modes, and if all three modes (RESET, OPERATE, and HOLD) are provided, then two relays, say X and Y, are required for each integrator (see Fig. 4.1.5). All X relays and Y relays are connected in parallel and the X and Y lines are wired to the control panel, where a *mode switch* controls the voltage applied to these lines (Fig. 4.1.6). The integrator control relays must have matched timing, i.e., all relays pull in or drop out at essentially the same time (see also Fig. 4.7.10). D-C relays are preferred to a-c relays because their timing can be matched more accurately, because they introduce less noise into the computer amplifiers, and because they are more efficient (less power required) than a-c relays. Typical time specifications for quality low-cost d-c relays are 16 ± ½ msec pull-in time and 5 ± 1 msec drop-out time.

The *relay power supply* consists of a transformer, rectifier, and filter; regulation is not required. Relay ground currents must not flow through computer signal-ground connections.

The following additional precautions prevent the relay control circuits from introducing errors in computation:

1. To minimize noise pickup due to ripple in the relay supply, the control relays are deenergized in the OPERATE mode.
2. The control circuits are suppressed either individually at each relay, or more economically at the X and Y lines in the control panel. For this purpose, varistors are convenient and low in cost.
3. The two relays associated with each integrator must be controlled in the correct sequence. For example, when the computer goes from RESET to OPERATE in Fig. 4.1.5*a* then, if the X relay contacts open before the Y relay (A_2) opens, the initial condition will change. It increases if A_1 of X opens first; it decreases if A_2 of X opens first. One can remedy this situation by delaying the removal of the voltage from the X line for a few milliseconds (Fig. 4.1.6).

4.1.7. BALANCE CHECK (POT SET) Mode. This mode serves one or more of the following functions:

1. To adjust the balance of d-c amplifiers
2. To set the coefficient potentiometers
3. To put the computer into a standby condition

The usual procedures for the zero adjustment of d-c amplifiers (see also Sec. 4.7) require each amplifier to have zero input and a resistance feedback. For integrators the RESET mode satisfies these requirements provided the initial-condition voltage is removed.

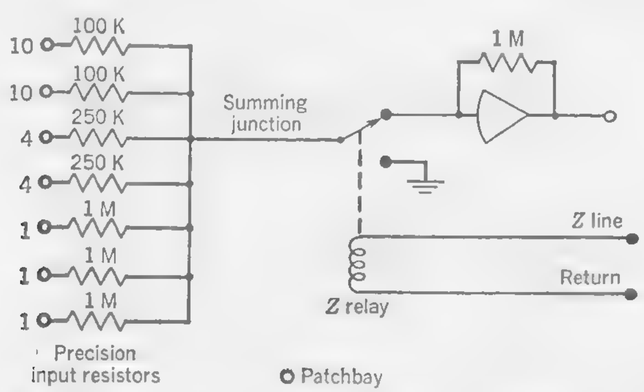


FIG. 4.1.7. Summing-amplifier relay circuit. Z relay is energized in BALANCE CHECK (POT SET) and deenergized in all other modes.

For each summing amplifier the inputs are removed by a relay, say Z, which disconnects the summing junction (SJ) from the grid (Fig. 4.1.7). All Z relays are connected in parallel, and the Z line is brought to the control panel. The three lines (X, Y, and Z) are connected to the relay supply in BALANCE CHECK (Fig. 4.1.6).

The relay contacts which disconnect the summing junction (SJ) from the grid can be used to ground the SJ, so that coefficient potentiometers connected to amplifier inputs will be correctly loaded (Figs. 4.1.5 and 4.1.7). The potentiometers can then be set accurately and without overloading amplifiers (see also Sec. 2.1).

If amplifiers are to be used in high gain, i.e., with feedback through special patchboard connections, then relays to add a feedback resistance in BALANCE CHECK are provided. An alternate method is to have a resistance feedback across each amplifier normally and to use a relay (when energized) to convert the amplifier to high gain. Then the converting relay is *deenergized* in BALANCE CHECK (Fig. 4.1.6).

In some computers, balancing is done in RESET with the aid of individual balancing switches on the amplifier panels.

In some computers the term STANDBY refers to removal of plate voltage; this permits the operator to leave the computer unattended. However, turning the plates ON and OFF frequently may increase the number of tube and capacitor failures because of large transients. Since in BALANCE CHECK all amplifier outputs are zero, this is a convenient standby condition, provided that the servos (and other electromechanical components) are also turned OFF. This may be accomplished by providing the servos with ON-OFF relays (ON with the relay energized).

4.1.8. REFERENCE OFF Mode. As the name implies, this mode removes the +100 volts and the −100 volts from the patchbay. This mode is required on computers with grounded metal patchboards to prevent the reference voltages from being accidentally shorted as the operator changes patchcords at the computer.

4.1.9. REPETITIVE Mode. This mode converts a real-time analog computer to repetitive operation, so that it cycles from RESET to OPERATE at a frequency of 1 to 30 cps. One of two methods may be used to perform this rapid switching. The first consists of replacing the regular X and Y relays with high-speed (1 msec operate time) relays and changing all integrating condensers from 1 to 0.1 μf. Then the X and Y lines are driven from a low-frequency oscillator at 1 to 15 cps. The output (solution) is displayed on a cathode-ray oscilloscope with a long-persistence screen. In this method the relay time delay will produce errors as large as 2 per cent at the higher frequencies, and if the repetitive mode is used continually the relays will wear rapidly. At 10 cps 30 hours of use require more than 1 million relay operations.

In the second method the X and Y relays are replaced by electronic switches that can be driven by an oscillator at 5 to 30 cps; a standard cathode-ray tube may be used. The integrating condensers are reduced from 1 to 0.01 μf (see also Sec. 6.2).

In the REPETITIVE mode electromechanical components such as servos cannot be used.

4.1.10. SLO-RESET Mode. This mode is the same as RESET for all electronic components and differs from it only in the special controls for electromechanical units such as servos. At the completion of any run the computer is switched from OPERATE or HOLD to RESET. The electronic components return to their initial value practically instantaneously (integrators usually have a time constant of 100 msec). Unless the servo position is near to its initial value, this switching to RESET produces a large step input to the servo and causes it to slew to the initial position. The servo slewing speed is much larger than that normally required during computation; so that the greatest contribution to potentiometer wear occurs during such slewing in RESET. Hence it is desirable to reduce the slewing speed during the initial RESET period, say by reduction of the power used to drive the servo motor for about 10 sec after switching to RESET (see also Sec. 3.1).

MISCELLANEOUS CONTROL FUNCTIONS

4.1.11. Control Panels. A centrally located control panel permits the operator to monitor and control the performance of all components. In addition to the mode switch one or more of the following may be supplied:

1. Amplifier selector (switch or push buttons)
2. Coefficient-potentiometer selector (switch or push buttons)
3. D-C voltmeter
4. Amplifier unbalance indicator
5. Master overload indicator
6. Calibration controls

The *amplifier and potentiometer selectors* enable the operator to monitor the outputs of these components without disturbing the patchboard. The arm of the amplifier selector is usually connected to the d-c voltmeter and to a point on the patchbay, the latter to permit read-out on external test equipment. The potentiometer selector arm is also connected to a patchbay terminal (see also Sec. 2.1).

The *d-c voltmeter* (preferably a vacuum-tube voltmeter) is used to make coarse (4 per cent of full scale) measurements in order to check relative magnitudes of computer voltages. Several ranges, such as 200, 100, 10, and 1 volt full scale, are needed.

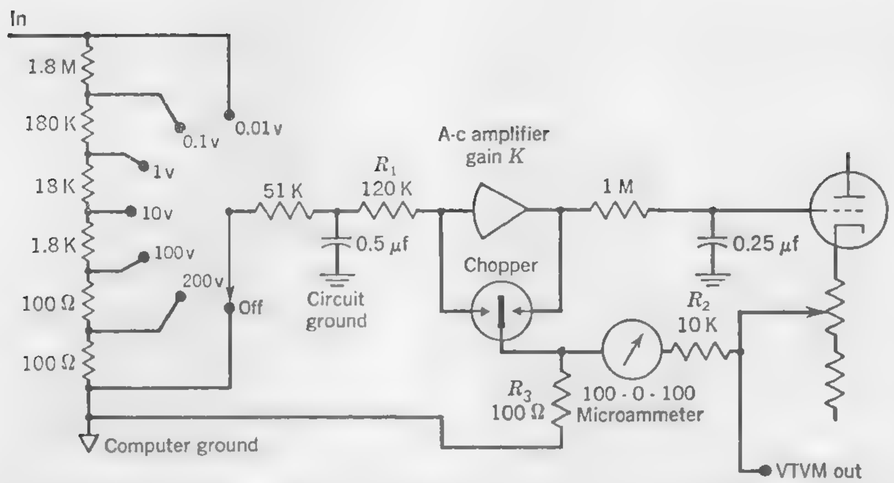


FIG. 4.1.8. Sensitive d-c vacuum-tube voltmeter with zero drift and high input impedance. The chopper is used for modulation and demodulation, and to compare the output voltage with the input. The high input impedance is due to series negative feedback. The input impedance is $R_{in} = 2R_1\left(\frac{KR_3}{R_2} + 1\right)$; for $K = 1{,}500$ one has $R_{in} = 3.8$ megohms. The voltage at the output terminal is zero if the meter current is zero, i.e., with zero input voltage (see also Fig. 4.1.9).

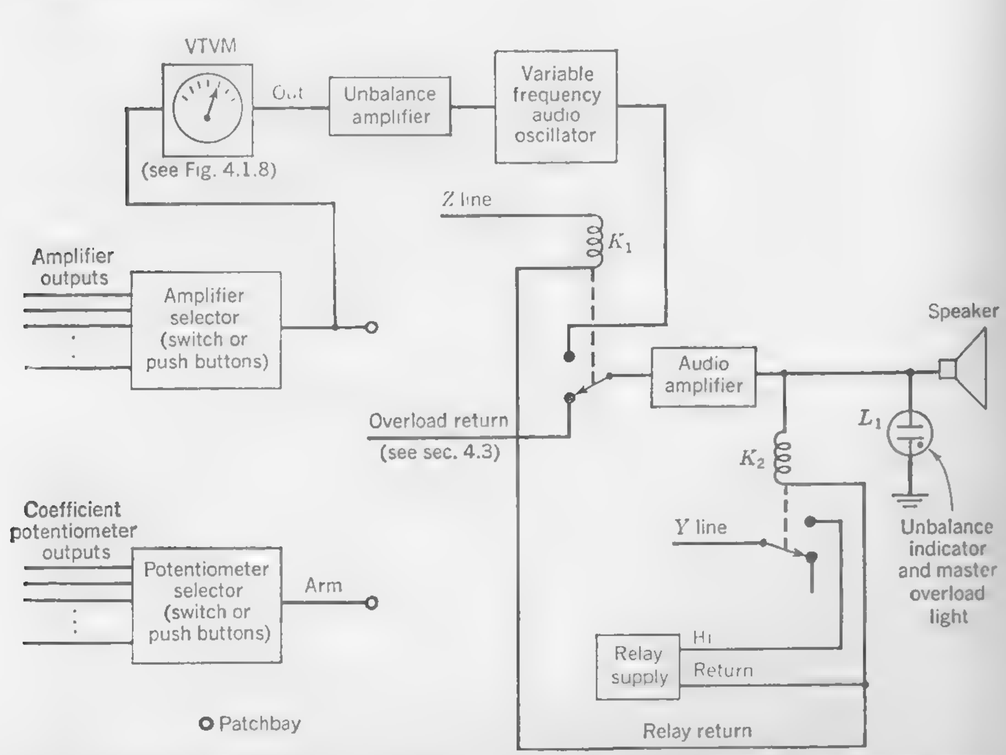


FIG. 4.1.9. Block diagram of control panel. Relay K_1 is energized in BALANCE CHECK. In this mode the neon light L_1 and the speaker serve as unbalance indicators; in the other modes they are overload indicators. If an overload occurs in OPERATE, relay K_2 switches the computer to HOLD. The mode switches normally located in the control panel are not shown in this drawing (see Fig. 4.1.6).

If an additional very sensitive range, such as 10 mv full scale, is included, the voltmeter may be used for accurate measurements using bridge or null methods (Fig. 4.1.8). The sensitive meter range can also be used for zero adjustment of d-c amplifiers; an unbalance as small as 100 μv can be detected.

To permit zero adjustment of amplifiers remotely located from the control panel an audible and/or visual (indicator light) *unbalance indication* can be provided.

The *master overload indicator* (see also Sec. 4.3) usually provides audible and visual signals to warn the operator of overloads. The master overload circuit may also be used to switch the computer to HOLD to permit the operator to locate overloads. A block diagram of a control panel comprising all the preceding circuits is shown in Fig. 4.1.9.

Calibration controls to set function generators, to balance electronic multipliers, etc., may be included in the control panel (see also Secs. 3.2 and 3.3). On the other hand, if these controls are located in the individual units, it is helpful to provide an indication at the control panel to warn the operator if any component is in "calibrate."

4.1.12. Slave (Parallel) Operation. If the problem is too large for a single computer, a second or even third computer may be added, provided that the following precautions are observed.

If only additional summing amplifiers, multipliers, resolvers, function generators are needed, i.e., no electronic or electromechanical (rate servos) integrators are used, the only requirement is that all computers use the same reference voltage. One reference supply with sufficient current capacity to drive all the computers can be provided. Alternatively, each computer may have a "slavable" reference supply, so that all of them may be slaved to a master unit.

If integrators of two or more computers are to be used in the same problem, then an additional requirement is that the integrator relays be controlled simultaneously. The simplest situation occurs if the integrators of two computers (A and B) use the same type of relay, and a single relay supply (large enough to drive all the relays) is used. Then corresponding relay lines are connected (X to X and Y to Y). The controls of the SLAVE UNIT B are set to that mode which deenergizes all relays (OPERATE in the system described above, Fig. 4.1.5). Station A is the *master*, and its control switch determines the mode of all integrators. Conversely, station B can become the master unit simply by switching station A to OPERATE.

If each computer has its own relay supply or different types (voltage or time-delay) of integrator relays are used, then additional control relays will be needed. The criterion for correct parallel operation of computers is that all integrators start and stop at the same time. It should also be remembered that difficulties are likely to be encountered if two (or more) relay supplies are operated in parallel. If the source impedances are not properly balanced, then the division of load may be so poor that one supply may have to drive the entire load.

4.1.13. Remote Control. It is desirable to provide additional control stations at several locations in a computer installation, so that the operator can change the computer mode without returning to the control panel; usually only RESET, OPERATE, and HOLD are needed. The principle of slave operation is used to provide these remote controls. A control station consists of a cable connected to the computer and a mode switch; the cable contains the X and Y lines and the relay-supply lines. Now the computer is switched to OPERATE (no voltage on the relay lines), and the remote-control switch is used to apply the relay voltage to obtain the selected integrator modes. If two or more remote-control stations are supplied, all except the master station are switched to OPERATE.

Output devices such as a recorder or a plotting board are locations at which it is convenient to place remote controls (see also Sec. 4.2). Some plotting boards are furnished with a mode switch, and the cable that carries the computer signals also has provision for the relay lines. Since the pen lift on such a plotter is also relay-operated, it is possible to use one pole of the mode switch to control the pen (down in OPERATE, up in RESET and HOLD).

4.1.14. Individually Controlled Integrators. A few individually controlled integrators may be required for special problems, such as the simulation of sampled-data

systems (see also Sec. 4.2). The III sides of the X and Y relays and the relay-supply lines are then made available at the patchbay (see also Sec. 4.6).

PART 2: SEMIAUTOMATIC PROGRAMMING

By BERNARD D. LOVEMAN

INTRODUCTION

4.2.1. Survey. Semiautomatic programming techniques for electronic analog computers may be divided into two categories. The first consists of specialized equipment which relieves the operator of repeated routine operations in studies requiring many parameter changes (parameter-optimization and component-tolerance studies). The parameters to be changed include coefficients, initial conditions, and nonlinear functions. The response of the system to different kinds of inputs may be investigated; these may include a step, a constant velocity, an exponential, or random noise. Since the equipment required is closely related to the task to be performed, many different automatic programmers have been constructed. Three types that are of general interest have been selected for detailed description. These are an autocycler, an output scanner, and an input-output programmer. In addition, one scheme for automatic recorder calibration is described.

The second category of semiautomatic programming techniques involves those special programs which permit a reduction in the number of runs required to solve a problem. These include circuits for determining the values of parameters which yield optimum results and the use of integrators as analog storage units (see also Secs. 5.11 and 6.6).

AUTOMATIC PROGRAMMING EQUIPMENT

4.2.2. Autocycler. The *autocycler* permits continuous operation of the computer with only nominal supervision by the operator. It may also be used to turn other equipment, such as recorders, on and off at specified times. The unit is particularly useful when many repetitive runs are desired.

The autocycler has three modes of operation:

Mode 1. *Fixed Automatic Cycling.* In this mode the periods of time assigned to the OPERATE (COMPUTE), HOLD, and RESET phases are independently adjustable and may be preset by the operator over a range of several minutes, typically zero to three.

Mode 2. *Variable Automatic Cycling.* In this mode the OPERATE phase is terminated by a signal (voltage level) from the computer. The duration of the OPERATE phase is not restricted.

Mode 3. *Manual.* In this mode the autocycler is used as a remote station for manual operation of the computer. Controls and indicator lights are provided to indicate various functions. An electric counter records the total number of OPERATE cycles in modes 1 and 2.

In Fig. 4.2.1 the autocycler is shown switched to mode 1. The computer is assumed to have the same integrator-relay operation as that shown in Fig. 4.1.5*a*. The computer mode switch is in OPERATE (no voltage on the relay lines). In mode 2 switch S_1 is reversed and the OPERATE phase is terminated by energizing relay K_1 with a signal from the computer on the external-connection line. Thereafter the cycle is the same as in mode 1. Remote-control operation (mode 3) is described in Art. 4.1.13.

By the addition of potentiometers, relays, and/or stepping switches it is easily feasible to extend the capabilities of the autocycler so as to permit changing coefficients between runs.

4.2.3. Output Scanner. The purpose of a *scanning system* is to record automatically and in sequence all amplifier output voltages in RESET and HOLD and all potentiometer coefficients in BALANCE CHECK (see also Secs. 2.1 and 4.1). A typical scanner can

accommodate as many as 200 different outputs. The principal components are a printer, an analog-to-digital converter, a sequencer, an output selector, a relay unit, and a control panel. Functions and typical specifications of these components are listed in Table 4.2.1.

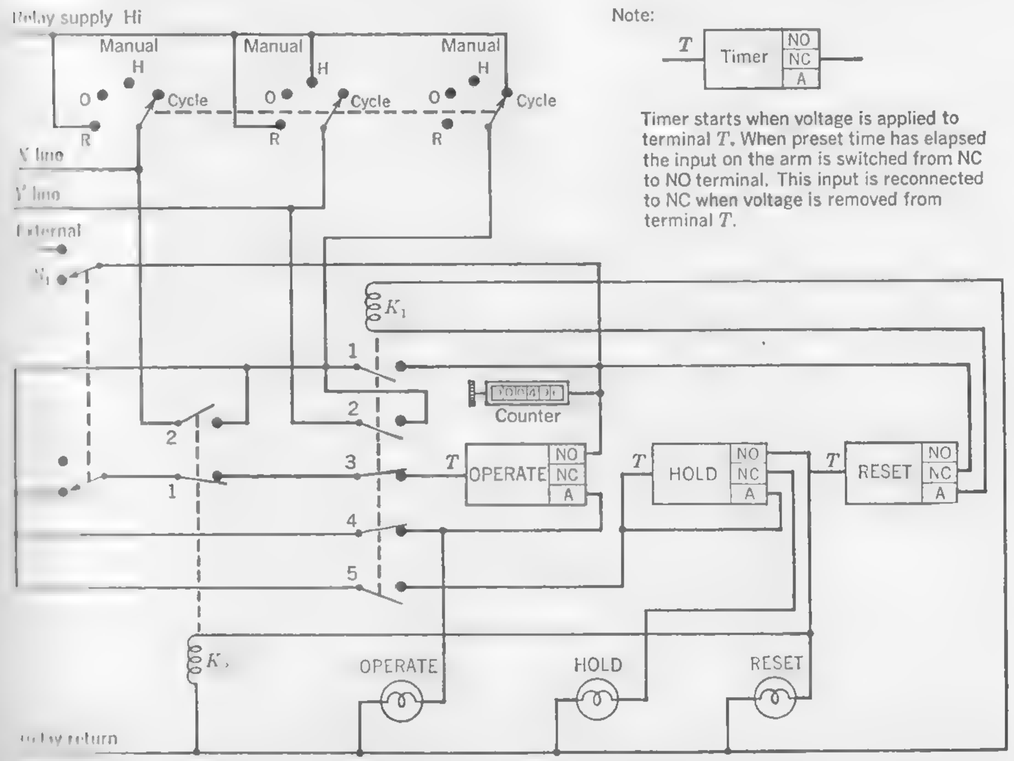


FIG. 4.2.1. Schematic of autocycler. Computer is switched to OPERATE (no voltage on the relay lines). With the switches as shown the autocycler is in mode 1. Operation is as follows:

Phase 1. OPERATE timer energized through contact 3 of relay K_1 and contact 1 of relay K_2. OPERATE light ON through contact 4 of relay K_1. X line and Y line deenergized. Computer in OPERATE.

Phase 2. Preset OPERATE time elapses and timer contact A switched to NO. This energizes relay K_1 through contacts of RESET timer. The contacts of relay K_1: (1) latch K_1 through contacts of RESET timer, (2) energize Y line switching computer to HOLD, (3) remove voltage from OPERATE timer and light, (4) apply voltage to HOLD timer and light.

Phase 3. Preset HOLD time elapses and timer contact A switched to NO removing voltage from HOLD light and applying voltage to relay K_2 and RESET timer and light. A contact of relay K_2 applies voltage to X line, switching computer to RESET.

Phase 4. Preset RESET time elapses and timer disconnects A from NC and deenergizes relay K_1. Its contacts remove the voltage from: (1) Y line, (2) HOLD timer, (3) RESET timer and light, (4) relay K_2. The contacts of relay K_2 remove the voltage from the X line, switching the computer to OPERATE, and apply voltage to the OPERATE timer. This starts the next cycle, and the run counter advances one step.

4.2.4. Input-output Programmer. This is a flexible digital input-output unit which accepts input data and transmits them to the analog computer; in turn, it receives and records the answers from the computer. It is used to carry out the following operations (Fig. 4.2.3):

1. To set automatically coefficient, initial-condition, or problem-check potentiometers and to verify the results
2. To measure the potentiometer coefficients (with 100 volts on the III side) and the potentiometer output voltages (III side connected to the problem)

3. To set up function generators from coordinate information and to verify the function
4. To read outputs in any order in response to input commands
5. To read outputs in sequence (scanning)
6. To control the computer modes for programming a set of runs

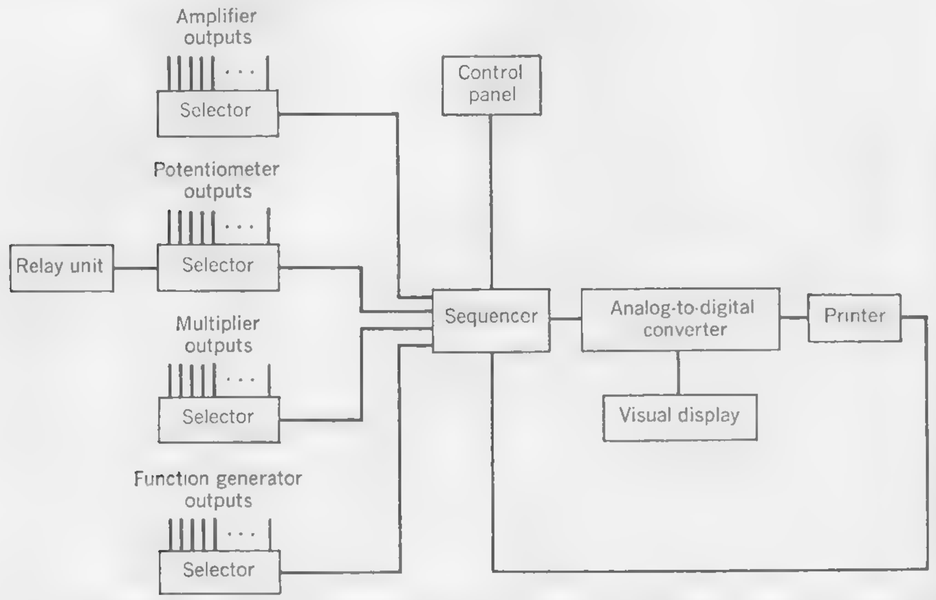


FIG. 4.2.2. Block diagram of output scanner system. Amplifier output voltages are recorded in RESET or HOLD; potentiometer coefficients in BALANCE CHECK.

Table 4.2.1. Characteristics of Output-scanner Components (see Fig. 4.2.2)

Component	Function	Typical specifications
Printer	Records simultaneously on one line the address, in a suitable code, of the component and its output voltage including polarity	11 characters per line, 100 lines per min
Analog-to-digital converter	Usually a digital voltmeter with a visual display. Its output is connected, after balance, to the printer	Voltage error as small as 10 mv. Balance speed 1 sec. Input impedance essentially infinite
Sequencer	Control center of scanner—generates address signal to printer	Time to advance one step 16 msec
Selector	Selects the output that is connected to digital voltmeter and consists of one or more of the following: relays, stepping switches, crossbar switches	Selects one of 200 different outputs in approximately 40 msec
Relay unit	Applies reference voltage (100 volts) to the coefficient potentiometers (see Sec. 2.1)	
Control panel	Used to start and stop scanning and to select starting point for scanning by component group such as integrators, summing amplifiers, function generators, multipliers, etc.	

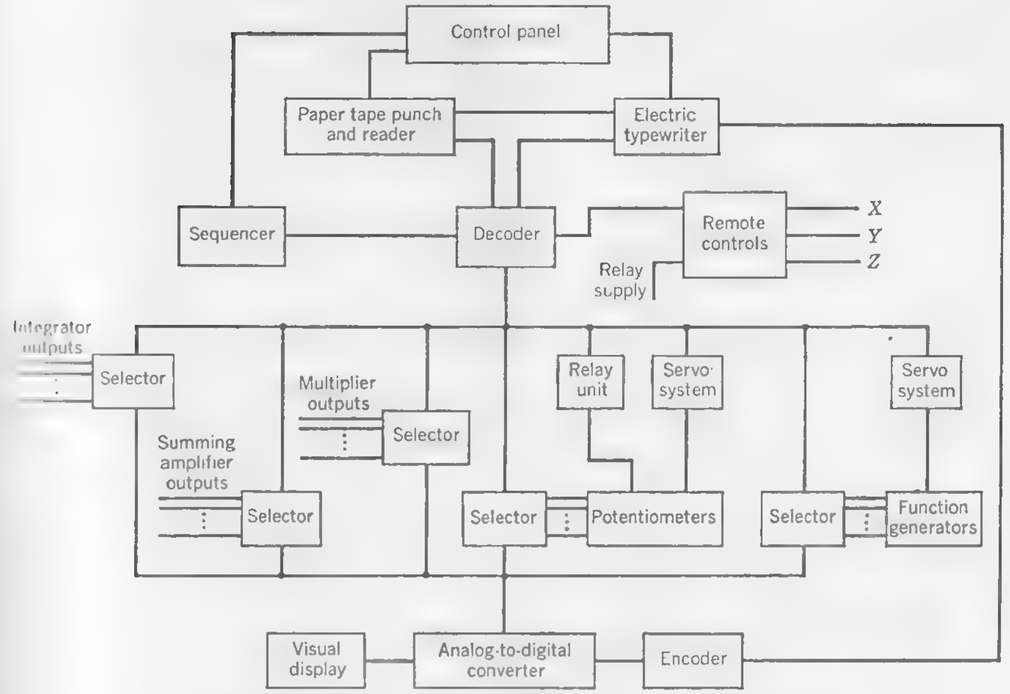


FIG. 4.2.3. Block diagram of input-output programmer (see text).

Some of the units used are described elsewhere in this handbook, namely, servo-set potentiometers (Sec. 2.1), servo-set function generators (Sec. 3.3), and sequencer and selectors (Art. 4.2.3). The remaining principal components and their functions are described below:

1. The *input units* consist of a punched-paper-tape reader and an electric typewriter or keyboard. Typical input or output speed is 8 characters per second. The typed copy is prepared in a suggestive code, e.g., spt 27 + 2347 for "set coefficient potentiometer number 27 to 0.2347."
2. The *output units* consist of a paper-tape punch and an electric typewriter. In some installations the input and output units are combined in one assembly (electric typewriter with a paper-tape punch and reader).
3. The *decoder* converts the input signals from serial pulses to a parallel-coded instruction and stores the command until it is executed (see also Sec. 6.5).
4. The *encoder* converts parallel output information from the digital voltmeter into serial coded signals that are printed out by the typewriter.
5. The *remote-control unit* provides relay switching to control the computer mode (Sec. 4.1) from typed commands or properly sequenced prepunched paper tape. This feature permits the operator to prepare a complete automatic program, with the end of the OPERATE phase controlled by a relay comparator circuit (Sec. 3.3). Other controls include a time delay after RESET and a circuit to prevent the computer from switching to OPERATE if any component is overloaded.

The punched-paper-tape input of this programmer contributes to a large extent to its versatility. Three fundamental computer operations can be carried out automatically, viz.,

1. Variation of parameter studies
2. Problem-checking procedures (Sec. 4.3)
3. Automatic equipment checks (Sec. 4.7)

4.2.5. Recorder Calibration and Controls. Multichannel variable-speed direct-writing oscillographs (strip-chart recorders) are used extensively as analog-computer output devices. Each computer run requires one to record run designation, paper

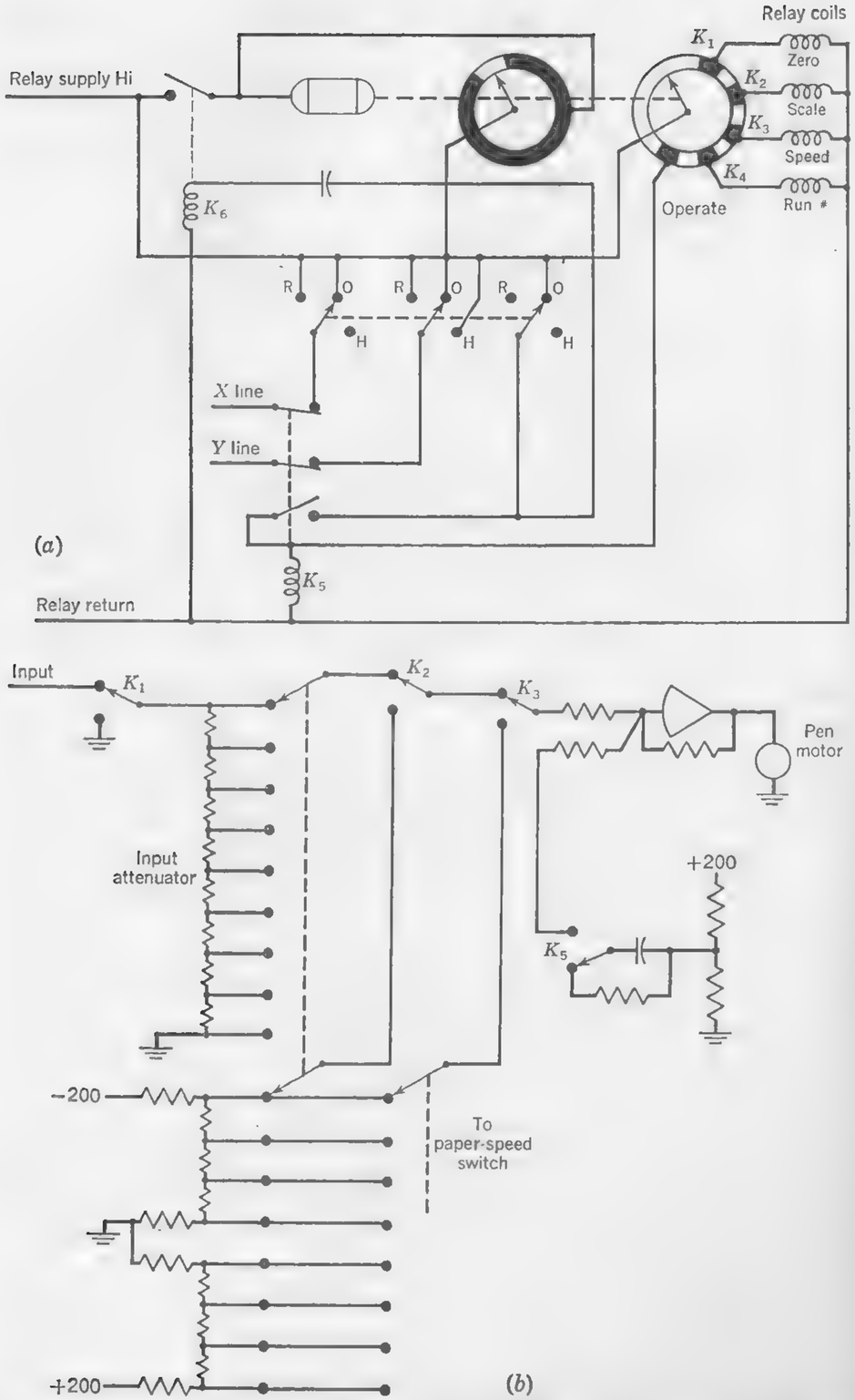


FIG. 4.2.4. Recorder calibration and controls. (*a*) Calibration sequence is controlled by motor-driven switch. Before each run, coded calibration deflections are recorded at a fixed paper speed. Then the paper speed previously set on the speed switch is resumed, and the computer is switched to OPERATE. (*b*) Switching circuitry for one channel.

speed, and recorder zero and scale for each channel; in larger computers this is often done automatically. One method uses relays and a motor-driven switch (Fig. 4.2.4). The calibration is performed at a fixed speed independent of the paper speed used in the problem. A useful run-designation unit can be constructed with a stamp and a counter operated by a solenoid before each computer run.

In addition, it is convenient to provide remote controls (Sec. 4.1) at the recorder control panel. After the mode control is switched from RESET to OPERATE, the computer remains in RESET until the calibration has been completed and sufficient time has elapsed for the paper to return to the correct speed. A pip or mark denoting the start of each run is an added convenience.

AUTOMATIC PROGRAMMING TECHNIQUES USING ANALOG STORAGE

4.2.6. Analog Storage Circuits. Automated parameter-optimization studies (Sec. 5.11), statistical studies (Sec. 6.3), and solutions of boundary-value problems (Sec. 5.9) and difference equations (Sec. 6.6) require circuits capable of storing analog-voltage values for insertion into the computation or into a later computer run.

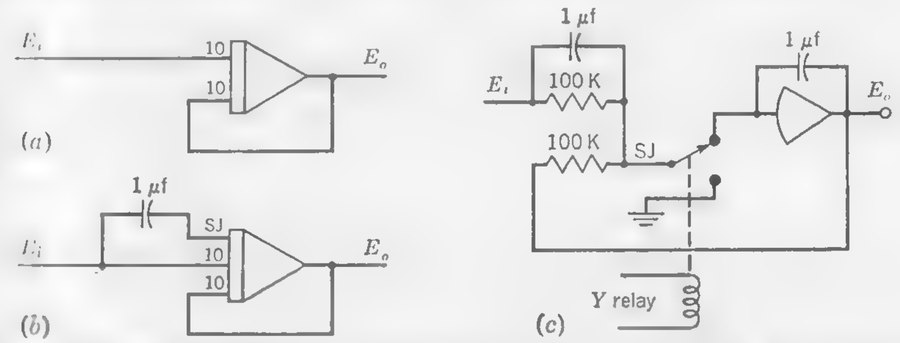


FIG. 4.2.5. Storage circuits. A standard integrator with individually controlled relays stores a signal whose derivative need not be available. These circuits have two modes: STORE and HOLD; in the first, the relays are deenergized and in the second, the Y relay is energized (Sec. 4.1). (*a*) Basic circuit. The disadvantage of this circuit is that the voltage stored will be in error because of the lag due to the RC time constant (typically 0.1 sec). In STORE, the transfer function is $E_o/E_i = -1/(1 + 0.1p)$. (*b*) Improved circuit. The input time constant is matched to the feedback; the transfer function in STORE is -1. (*c*) Detailed schematic of circuit *b*.

Every d-c integrator will store its output voltage in the HOLD condition (Sec. 4.1). Individual integrators can be switched to HOLD if their hold relays can be controlled through the computer patchboard (either individually or in groups). Figure 4.2.5 shows how a conventional integrator is connected to store a voltage whose derivative may not be available in the computer setup; this is the basic analog storage circuit. A disadvantage of integrator storage is that it does not remain accurate for an indefinite time; this disadvantage can be overcome through combination of integrator storage circuits with servo-set potentiometers.

High-speed sampling/storage circuits and their applications are discussed in Secs. 5.3, 5.4, and 6.3.

4.2.7. Automatic Scale Changing. It is frequently necessary to study a system whose variables have large dynamic ranges. One approach is to divide the problem into two phases with different voltage scales for these variables in each phase. Storage circuits are used to HOLD the integrator outputs at the end of the first phase; these voltages are then used as the initial-conditions for the second phase with rescaling as needed. In order to implement the scale change automatically, another relay, say W, is needed for the initial-condition circuit of each integrator; in the second phase, a new mode called RESET II is required. The mode sequence in the first phase is the usual RESET, OPERATE, and HOLD; in the second phase RESET II (initial conditions

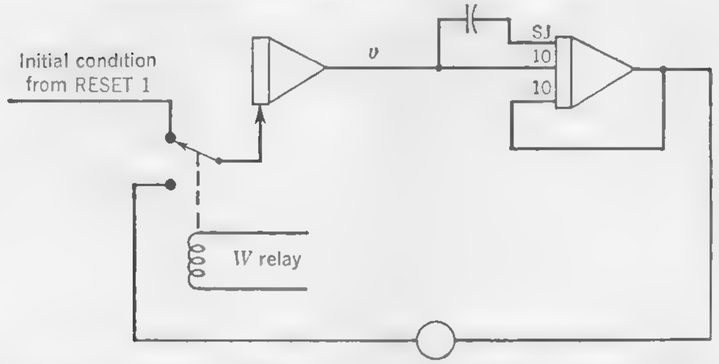


FIG. 4.2.6. Voltage-scale change. The voltage v is stored when the integrator is in OPERATE. The store unit is in HOLD when the integrator is in RESET I, RESET II, or HOLD. The stored voltage with desired scale change is used as the initial condition in RESET II. When the scale of the voltage v is changed (phase II), then all computer circuits fed by v must be suitably adjusted (see text).

from the storage circuits), OPERATE, and HOLD (Fig. 4.2.6). Other methods for programming variables with a large dynamic range include transformations (e.g., logarithmic transformation) which reduce this range and automatic continuous scale change (Sec. 5.5).

References (see also Secs. 5.11, 6.3, and 6.6)

1. Bennett, R. R., and A. S. Fulton: Automatic REAC Operation for Statistical Studies, *Project Cyclone Symposium II*, Reeves Instrument Corp., New York, 1952.
2. Castanias, R. P., and J. E. Sherman: Review of Computer Progress in 1957, *IRE Trans. on Electronic Computers*, March, 1958.
3. Follin, J. W., G. F. Emch, and F. M. Walters: Modifications and Additions to the REAC, *Project Cyclone Symposium II*, Reeves Instrument Corp., New York, 1952.
4. Hamer, Howard: An Automatic Digital Input-output System for Analog Computers, *Proc. First Flight Simulation Symposium*, September, 1957.
5. Harris, Holton E.: New Techniques for Analog Computation, *Instruments and Automation*, May, 1957.
6. Johnson, Clarence L.: *Analog Computer Techniques*, sec. 10-8, McGraw-Hill Book Company, Inc., New York, 1956.
7. Weil, Herschel: Reduction of Runs in Multi-parameter Computations, *J. Assoc. Computing Machinery*, April, 1955.

PART 3: PROBLEM-CHECKING CIRCUITS AND PROCEDURES

By BERNARD D. LOVEMAN

INTRODUCTION

4.3.1. Survey. Assuming a complete and correct mathematical problem statement and setup diagram, checking procedures include

Routine equipment checks (Sec. 4.7)

Programming checks (patchboard verification, static checks, integrating-capacitor check, time-scale check) (Arts. 4.3.5 to 4.3.8)

Monitoring of excessive and insufficient signals and parasitic oscillations during computer runs (Arts. 4.3.2 to 4.3.4)

Solution checks ("reasonableness" and repeatability of solution; comparison with numerical or digital-computer check solution; check solutions for special parameter values; checking by resubstitution into differential equations)

Equipment-test acceptance accuracies are guided by manufacturer's specifications, but the accuracy obtained in programming and solution checks depends on the problem. Experience shows that a large error (say 0.5 per cent in a computer with 0.1 per cent components) in a static check is usually due to an erroneous setup or a component failure rather than to error accumulation. If a large error or lack of precision (repeatability) due to accumulation of small errors occurs, then reprogramming or a more accurate computer is required.

EXCESSIVE AND INSUFFICIENT SIGNALS

4.3.2. Amplifier Overload Circuits. D-C amplifier output voltage exceeding rated limits (say ±100 volts) can operate neon overload lights or relay-actuated visual

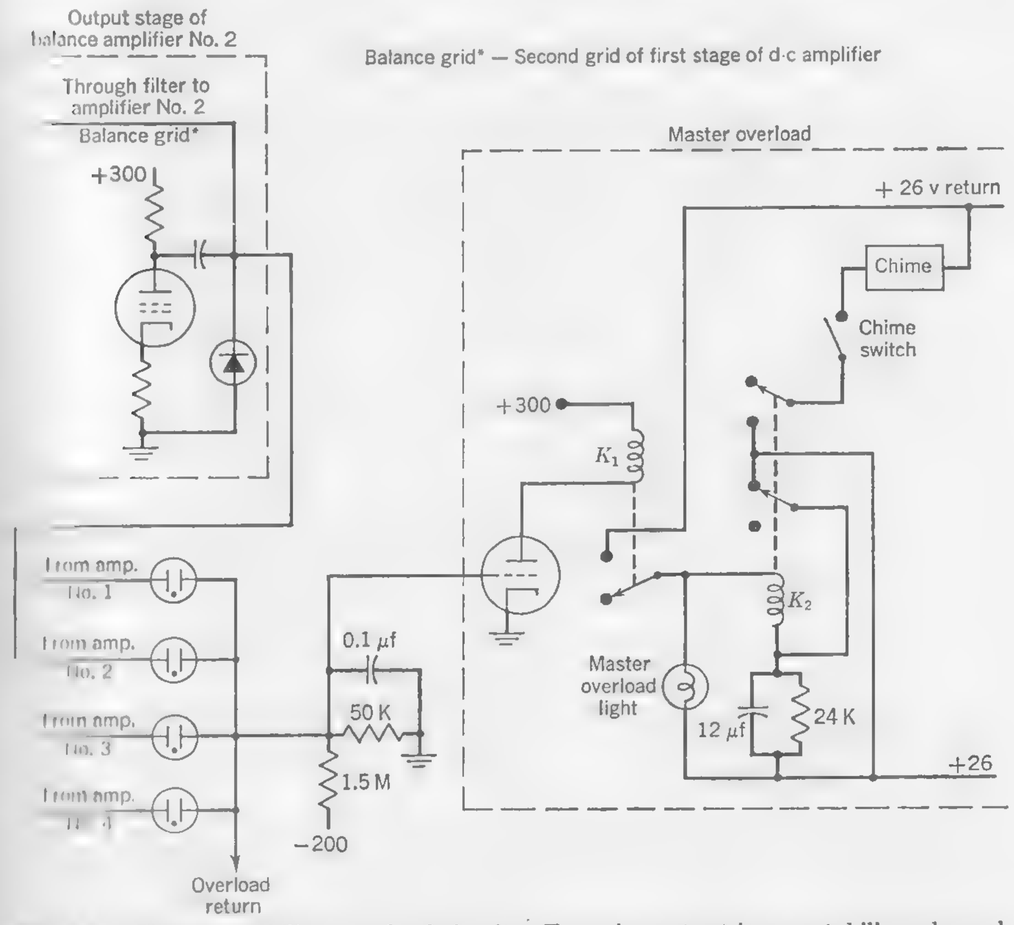


FIG. 4.3.1. Chopper-stabilizer overload circuit. Excessive output in any stabilizer channel (Fig. 2.3) lights a neon bulb and actuates relay K_1; intermittent operation of relay K_2 rings the chime.

and/or audible overload indicators. In modern chopper- or commutator-stabilized d-c amplifiers (Sec. 2.3) overload circuits monitor the stabilizer-channel output rather than the amplifier output (Fig. 4.3.1). *Such circuits indicate nonlinear amplifier operation due to excessive output current and many amplifier malfunctions as well as excessive output voltage;* and they will admit output voltages in excess of nominal overload if the amplifier still operates linearly (say ±120 volts output with a light load!). Since a momentary overload indication might be overlooked, *stored-overload circuits* keep the overload light lit until extinguished by a push button or by the computer

reset switch (Fig. 4.3.2). *Overload-hold systems* place the computer in HOLD (Art. 4.1.5) whenever an overload occurs to permit convenient investigation of scaling, etc.; the overload-hold feature can usually be turned off at the operator's option.

Overloads in electronic multipliers and function generators are monitored in the d-c amplifiers associated with them.

4.3.3. Computer-servo Overload Indicators. Servo overload circuits similarly monitor servo-amplifier output voltages or motor control-field current to indicate

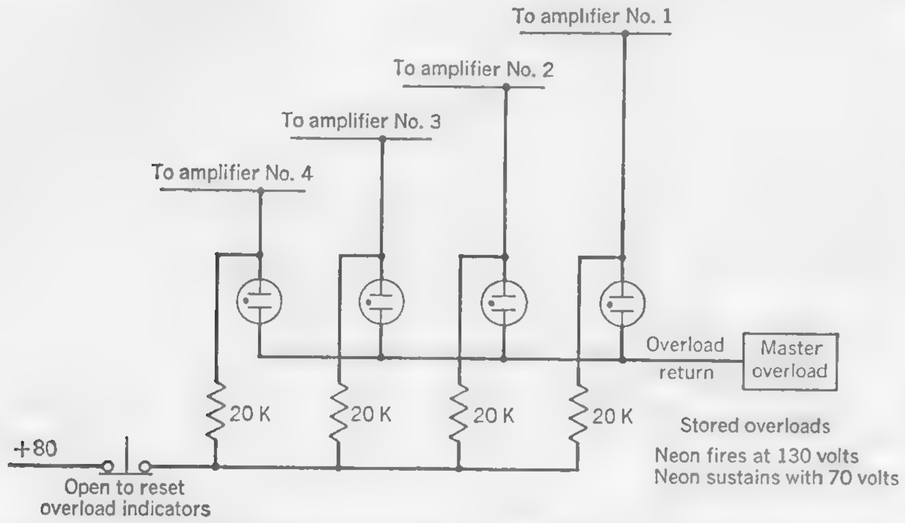


FIG. 4.3.2. A simple stored-overload circuit. Amplifier connections are to chopper-stabilizer channels (Sec. 2.3).

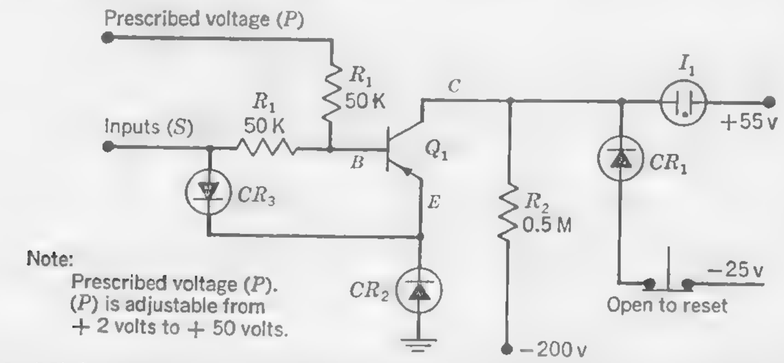


FIG. 4.3.3. Minimum-excursion indicator circuit. Neon indicator I_1 is turned on at the start of every run. Light is extinguished and remains off if the magnitude of input $|S|$ exceeds prescribed voltage P any time during the run (see text).

excessive servo *error*, and hence lack of dynamic response and malfunctions as well as overtravel. Servo overload circuits may be connected to the master overload system as d-c amplifiers, or they may operate a separate master overload indicator.

4.3.4. Minimum-excursion Indicators. A minimum-excursion indicator (MEI) indicates that a computer voltage S has exceeded a prescribed minimum voltage P during a computer run as a *check on computer scaling*. Figure 4.3.3 shows a Reeves transistor MEI circuit (U.S. Patent 2,851,638). MEI lights for all computing elements are lit at the beginning of each computer run; scaling is considered satisfactory if all MEI lights go out during the run and no overloads are observed. If an MEI remains on, its computing element is checked for rescaling or for a component failure.

Referring to Fig. 4.3.3, transistor Q_1 will not conduct unless its emitter-to-base voltage exceeds 0.1 volt. Neon I_1 (Ne 97) fires at 130 volts and sustains at 70 volts.

When Q_1 is not conducting, indicator I_1 can be ignited by the reset button (applying 200 volts through R_2) and remains illuminated because −50 volts is supplied to terminal C via CR_1 when the button is released. This condition continues as long as the magnitude of the input remains less than the prescribed voltage, i.e., $|S| < P$.

Assume the signal S is *negative;* then diode CR_3 does not conduct, and the emitter potential is zero. Transistor Q_1 will begin to conduct when $(S + P)/2$ reduces to approximately −0.1 volt, i.e., $|S| = P$. Conduction of Q_1 reduces the potential across I_1 to below 70 volts, and the neon is extinguished. Indicator I_1 now remains in this condition regardless of the input signal because terminal C is clamped at −50 volts, and 90 volts is not enough to fire the neon.

If the signal is *positive*, diode CR_3 conducts and the potential of the emitter is S. Therefore the emitter-to-base voltage is $(S - P)/2$. As a result, the transistor conducts when $S = P$.

PROGRAMMING CHECKS

4.3.5. Patchboard Verifiers. In most computers the patchboard connections are checked by visual examination. The patchboard verifier (PBV) performs this check automatically and rapidly; it produces a printed list which itemizes all patchcord

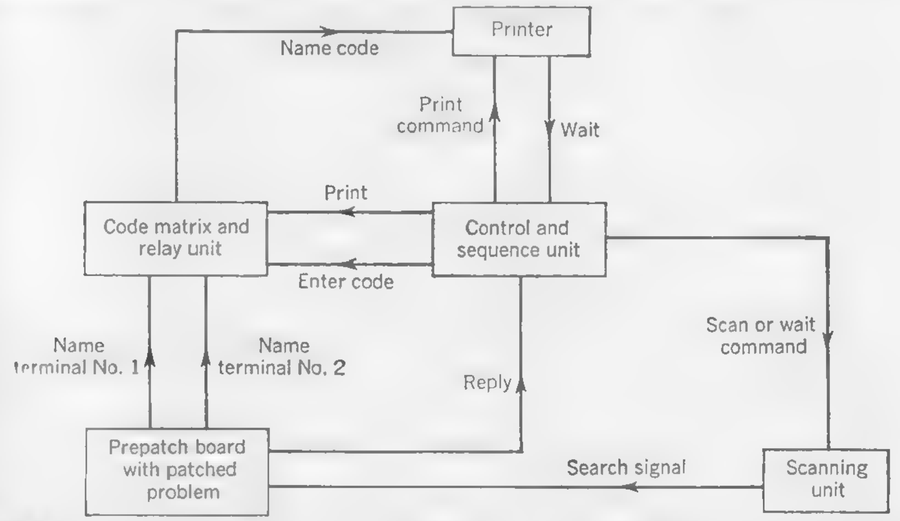


FIG. 4.3.4. Patchboard verifier—block diagram. Every patchboard terminal is scanned in turn to determine whether a patchcord is connected to it. When a patchcord is detected the scanning stops and the printer records both the source terminal and the destination terminal.

connections in pairs (connected terminals). A block diagram of the PBV is shown in Fig. 4.3.4. The basic operation is as follows:

1. Every patchboard terminal is interrogated in a prescribed sequence by applying a voltage to the terminal.
2. At the same time that the voltage is applied to the terminal, all succeeding terminals are examined to ascertain if a patchcord is connected to the terminal under interrogation.
3. If there is no patchcord, the next terminal is checked, etc., until a patchcord is detected.
4. When a patchcord is detected, the scanning stops until the printer records both the source terminal and the destination terminal of the patchcord.
5. The scanning then resumes until the next patchcord is found, etc.

Figure 4.3.5 illustrates one method for performing this operation. $P, Q, R, S, \ldots$ denote the patchboard terminals. When patchboard terminal P is interrogated, the arms SW_1, SW_2, and SW_3 of the scanning stepping switch are connected, respectively,

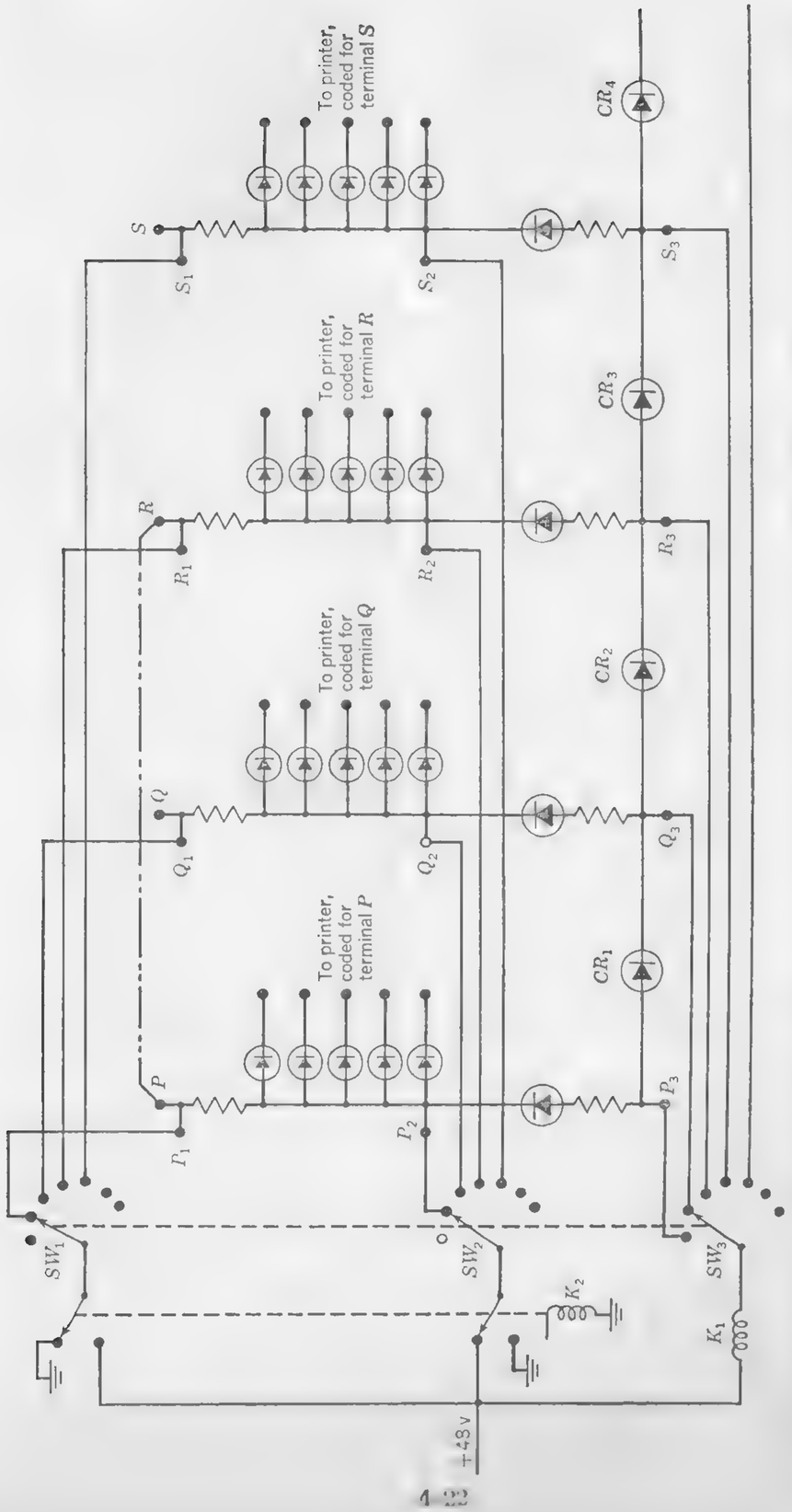


FIG. 4.3.5. Patchboard verifier—schematic diagram. The scanning stepping switch (decks SW_1, SW_2, and SW_3) is shown checking terminal P. A patchcord is connected between terminals P and R, shown as a dot-dash line, and consequently relay K_1 is energized through diode CR_2. Closing this relay halts the scanning and the designations for terminals P (say INT 17 OUT) and R (say POT 36 HI) are printed. The diodes CR_1, CR_2, CR_3, etc., prevent patchcord connections from being recorded twice.

to P_1, P_2, and Q_3. At the next position the stepper arms are connected to Q_1, Q_2, and R_3, etc. Assume the stepper is checking terminal P and a patchcord (shown dotted) is connected between P and R. In this position ground is connected to P_1, +48 volts to P_2, and the low side of relay K_1 to Q_3. Because of the patchcord K_1 is energized, and

1. The stepping switch stops.
2. +48 volts is connected to the printer coding diodes and the designation of terminal P (say INT 17 OUT) is printed.
3. Relay K_2 is energized, and +48 volts is connected to the printer coding diodes of R and the designation of terminal R (say POT 36 HI) is printed. K_2 is deenergized. Relay K_1 is also deenergized when the printing is completed, and the stepping switch continues to the next position, i.e., to interrogate terminal Q. Note that no printing takes place when terminal R is reached because K_1 is not energized. Thus connections for each patchcord are recorded only once. No multiple patchcords or portable multiples may be used.

Other special features included in the PBV are:

1. It will detect and indicate faults (such as a short circuit between a patchcord and its shield) from any terminal to the ground frame of the portable patchboard.
2. The designation for each patchcord terminal is printed in a code suggestive of the computing element, e.g., POT 36 ARM. In addition, each terminal is assigned a numerical value; subtotals and a total are printed out. If the problem is patched more than once, this feature permits a rapid check (without examining each printed line) of the later patchings.
3. A SELF CHECK mode is provided. The PBV prints in succession the code for every terminal on the patchboard. In this way the functioning of all stepping switches and printer coding units can be verified.

4.3.6. Static Checks. A static check is made in RESET and consists of applying initial conditions to one or more integrators and reading the resulting voltage outputs of all other amplifiers, multipliers, and function generators. The values of the output voltages are also computed from the equations; if the setup is correct, the computed values will agree with the measured voltages. This test will detect mistakes in patching and potentiometer settings as well as malfunctions in amplifiers, servos, function generators, etc. The test will not check computing resistors and potentiometers used at the inputs to integrators. Furthermore, in order to read nonzero voltages at every component, fictitious initial conditions may have to be used in the static test. This makes it possible for the wrong initial conditions to be used in the problem if the operator forgets to restore the problem initial conditions.

New computers have special integrator circuits for performing tests similar to static check, but without its limitations; connections to all other components remain the same as in RESET. For example, Problem Check* is shown schematically in Fig. 4.3.6. In PROBLEM CHECK STATIC the X and PC relays are energized. As a result the integrator shown is connected so that a feedback resistor parallels the integrating capacitor, and the amplifier output is disconnected from the INTEGRATOR OUT terminal on the patchbay. This output is replaced by a voltage from the Problem Check potentiometer which serves as an input to the problem. Furthermore, the Y relay is deenergized and hence the summing junction of the input resistors is connected to the amplifier. The steady-state output of the amplifier is therefore the negative of the sum of its inputs.

The Problem Check potentiometers introduce constant voltages into the problem; these in turn cause voltages to appear at the output terminals of each computing element. All output voltages are recorded and compared with theoretical values.

A very important feature of this circuit is that the checking procedure is carried out without disturbing the problem setup: no changes in patching, scale factors, or initial-condition potentiometer settings are required.

Another method for carrying out static checks is shown in Fig. 4.3.7. In this method initial conditions are used as inputs to the problem. Here the summing

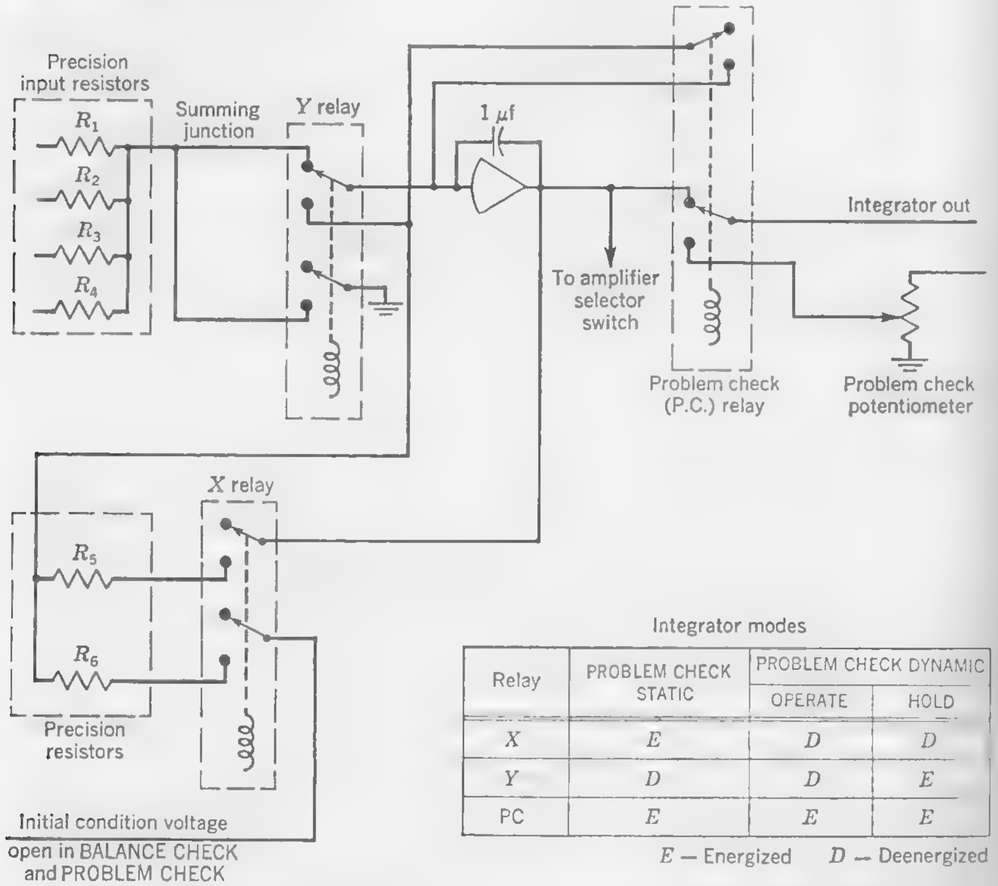
* Patent pending, Reeves Instrument Corp.

junction of each integrator can be connected in turn to a test amplifier, and thus the initial integrator rates may be checked.

4.3.7. Integrating-capacitor Tests. In PROBLEM CHECK DYNAMIC* (Fig. 4.3.6):

1. The X relay for each integrator is deenergized, and this disconnects its feedback resistor. This puts the integrator into OPERATE.

2. After a fixed time interval T determined by a precision timer, each Y relay is energized. This disconnects the summing junction from its amplifier and the integration stops.



Integrator modes

Relay	PROBLEM CHECK STATIC	PROBLEM CHECK DYNAMIC	
		OPERATE	HOLD
X	E	D	D
Y	D	D	E
PC	E	E	E

E — Energized D — Deenergized

FIG. 4.3.6. Problem-check circuit. Relay states for PROBLEM CHECK STATIC and PROBLEM CHECK DYNAMIC are given in the table (also see text).

The output of each integrator should now be $(1 + T)$ times its initial value. The output voltage of the other computing elements will meanwhile remain unchanged because the integrator outputs remain disconnected from the problem.

In the second method, called *capacitor test* (Fig. 4.3.8), each integrator is in OPERATE (zero initial conditions) with its input resistors disconnected. A constant voltage is supplied to each integrator through an additional precision resistor. After a time interval T the input voltage is disconnected and the output voltages of all integrators are measured. They should all be equal; any discrepancy is due to the integrating capacitor or, possibly, to the precision resistor.

4.3.8. Time-scale Checks. In most modern computers drift in integrators has been reduced to such a small value (as small as 0.1 volt in 90 min) that the major difficulty in choosing the proper time scale consists in avoiding excessive speed. The purpose of *time-scale check* is to determine whether the dynamic errors are excessive.

* Patent pending, Reeves Instrument Corp.

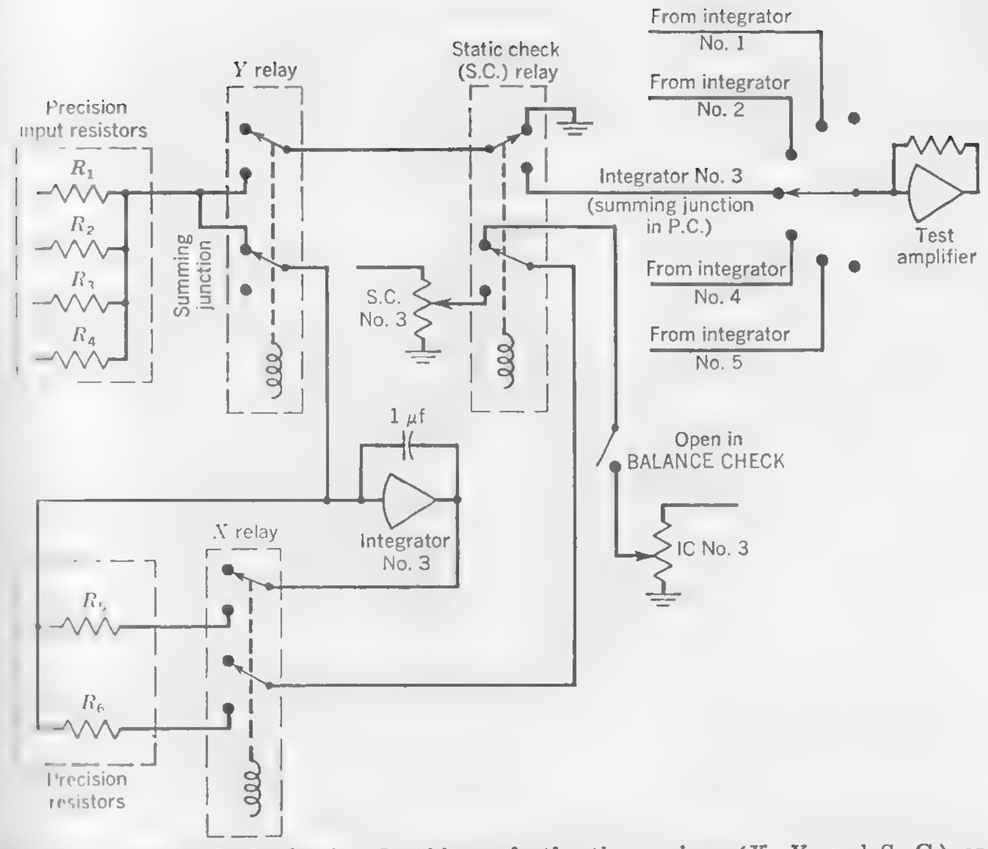


FIG. 4.3.7. Static-check circuit. In this mode the three relays (X, Y, and S. C.) are energized. The summing junction of integrator 3 is connected to the test amplifier, and its output is the initial rate; the output of integrator 3 is the static-check voltage.

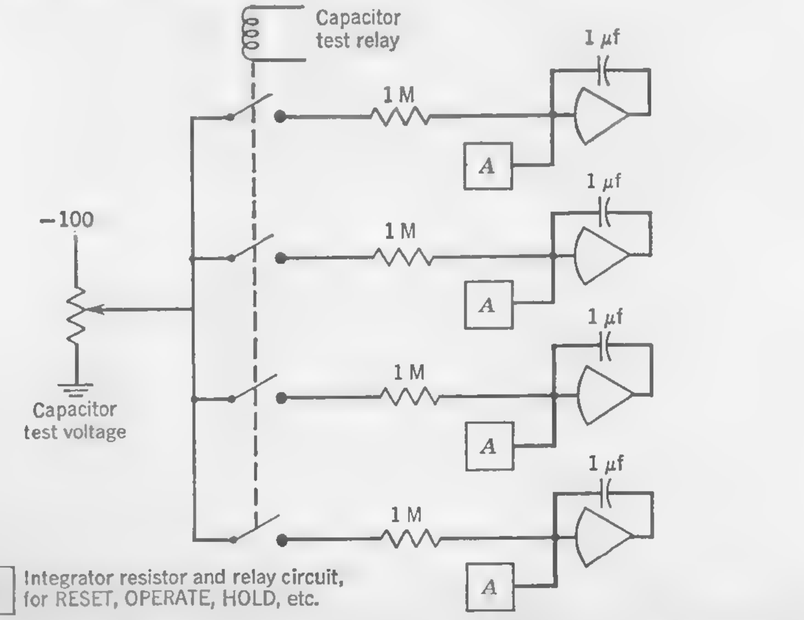


FIG. 4.3.8. Capacitor test. The capacitor test relay is energized for a time interval T, and during this period all integrator outputs increase linearly. If the time constants are equal, the outputs at time T will be equal.

In this mode the time scale of the problem is reduced automatically, typically to one-half the nominal value. This is accomplished by doubling all integrator feedback condensers. If the half-speed solution agrees with the original result, the dynamic errors are negligible, i.e., the original problem time scale is not too fast. Time-scale checks cannot be performed if the computer setup includes actual system hardware.

References

1. Johnson, C. L.: *Analog Computer Techniques*, chap. 11, McGraw-Hill Book Company, Inc., New York, 1956.
2. McCoy, R. D., and B. D. Loveman: Problem Check for Analog Computers, Reeves Instrument Corp., Reprint 102.
3. Meissinger, H. F.: Substitution Methods for the Verification of Analog Solutions, *Proc. Natl. Simulation Conf.*, 1956.
4. Richmond, W. F., and B. D. Loveman: Checking Analog Computer Solutions, *Project Cyclone Symposium II*, Reeves Instrument Corp., 1952.

PART 4: PATCHBAYS, INTERCONNECTIONS, AND ENCLOSURES

By THADDEUS J. KUSTO

PATCHBAYS

4.4.1. Introduction. Most multipurpose electronic analog computers are programmed by means of a patchboard system which comprises (1) a *patchbay* with spring-contact terminations for the computing elements and (2) interchangeable removable *problem boards* which carry the actual interconnecting *patchcords*. The programming of each problem depends on the operator's experience and ingenuity, since different valid computer setups can yield solutions of different accuracy. It is thus important to have a problem board which permits many useful interconnections with a minimum of confusion.

Electronic analog-computer setups involve closed loops of computing elements. Most of these are operational amplifiers operating on summing-point voltages between 10^{-3} and 10^{-6} volt over a working frequency range which rarely exceeds a few hundred cps for real-time simulation; but considerations of stability and dynamic-error reduction (Sec. 2.4) require amplifier frequency response up to several hundred kilocycles. The effects of shunting and coupling impedances must, then, be considered over the latter frequency range, since these impedances can become severe weighting factors on the accuracy and stability of a computer (Sec. 2.4). The design of an analog computer can be only as successful as the degree of consideration given to *low-level (summing-point) voltages, leakage between interconnections,* and *shunt and coupling impedances.*

4.4.2. Patchboard Layout. The number of patchcord connections that can be made on the problem board determines the number of components that can be terminated at a patching station and further determines the degree to which a computer can be expanded into a computer system. The number of patchholes that a problem board may be made to contain is determined by compromise between physical dimensions and required hole spacing. Since the legend identifying the component termination must be visible to the operator after patchcords are inserted, consideration must be given to the appearance and the utility of the board carrying a full problem program. In addition, the rear of the patchbay must be accessible for service and maintenance after it is fully wired. These considerations, together with mechanical requirements, usually dictate a hole capacity between 2,000 and 3,600, the actual number being determined by the computing capacity of the equipment.

Patchbay systems used on large analog computers can be classified into patchbays of *cellular (shielded)* and *molded* construction.

4.4.3. Cellular or Shielded Construction. The system shown in Fig. 4.4.1 incorporates all the features desirable in patchboard design and will serve to illustrate these

features. One of the prime requirements is *mechanical rigidity* to ensure unfailing achievement of contact between all patch tips and their connecting springs with every problem-board insertion. The patchboard of Fig. 4.4.1 is mounted in a rigid cast-aluminum frame with hardened-steel inserts at the working surfaces of the board and bay. Complete shielding is achieved by a method which leaves the front of the patchboard fully insulated while still permitting utilization of coaxial patchcords. An interlocking metallic grid surrounds nylon inserts designed so as to accept the patchcords with a locking action while providing a means of grounding the coaxial-shield-termination ferrules to the metallic grid by means of helical springs. Copper inserts are provided in four quadrants of the board, so that the metallic grid may be

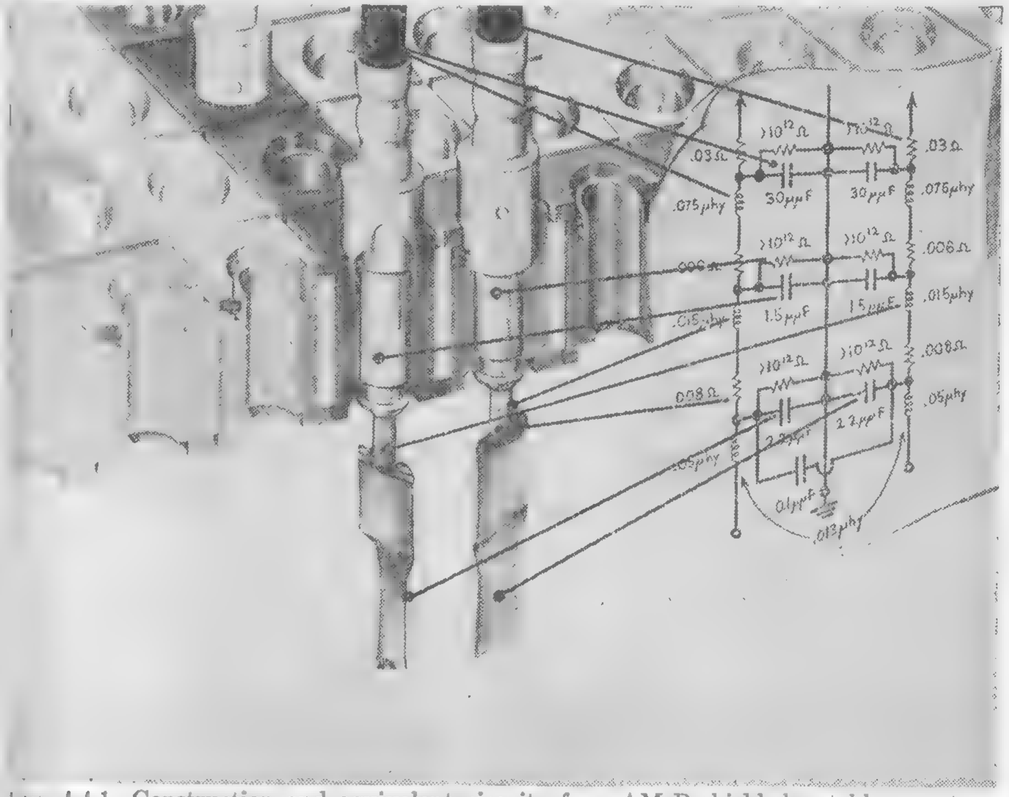


FIG. 4.4.1. Construction and equivalent circuit of an AM-P shielded patchbay system. Note that this construction grounds patchcord shields but does not permit accidental grounding of patchcord tips. (*Aircraft Marine Products, Harrisburg, Pa.*)

used as a unipotential ground system which connects directly to the computer ground system. This feature enhances hole utilization, since it permits one to ground computing components by means of a shorted patch tip in a single hole. Shielded patchcords obviate noise pickup and leakage and reduce capacitive coupling between patchcord connections to 1 $\mu\mu$f. Hole spacing is 0.375 in.; this constitutes a good compromise for the physical size of the board as related to the number of contained holes and legibility of patchhole identification. Recession of the metallic grid and the design of the patchcord tips effect a longer creepage path between coaxial center-conductor pins. Direct leakage between patchcord connections through the cross section of the board is prevented by the metallic grid which surrounds each nylon insert. The high insulation resistance and low capacitance achieved by this type of construction are illustrated by the equivalent circuit also shown in Fig. 4.4.1.

The patchbay is similarly constructed; its nylon inserts bear the contact springs which incorporate taper-pin receptacles. These receptacles permit the use of solderless wiring and the use of computing elements plugged directly into the contact springs

without intermediate wiring. Engagement between patchcord tips and the spring contacts of the bay is made with a wiping action which ensures a low-resistance contact under a spring pressure of 6 oz. Both spring contacts and patchtips are gold-plated to enhance conduction of low-level currents through a contact area which is rated at 5 amp per circuit to prevent damage or burnout due to faulty patchcord connections.

A more economical method of patchbay design utilizes a patchbay which incorporates many of the features described above but utilizes a *metal patchboard* in which the holes are drilled. Shielded patchcords are used with shield-termination ferrules grounded to the metal board. The economy of such design entails some minor disadvantages. Some flat panels, even when stress-relieved and reinforced, have a tendency to flex and lose contact alignment when patchcord plugs exert heaviest spring-contact pressure at the problem-board extremities. This condition may be alleviated by intelligent problem-board layout. The insulating finish on a metallic board is quickly scored by shield-ferrule contact, so that inadvertent shorting to ground of patchtips is not readily avoided during patching changes unless high voltage is removed from the patchbay for such changes. The design does, however, retain most of the features of the system shown in Fig. 4.4.1 at a substantial saving.

4.4.4. Molded Construction. Patchboard systems comprising phenolic patchboards and patchbays are adaptations of systems used with digital computers, where insulation and shielding characteristics are less important. Mechanical stability is achieved by means of a sturdy metal frame, but hole spacing is generally too close for optimum patchhole identification, so that holes may have to be sacrificed to prevent overcrowding of connections. Molded phenolics are characterized by high surface resistance, but cross-sectional leakage is comparatively high and unpredictable because of fillers used in the molding process. It is impossible to avoid leakage between connections, particularly if the patchtips contact the cross-sectional area of the board.

Table 4.4.1. Patchbay Accommodations for Two Analog Computers

Component description	Company *A*		Company *B*		Company *A*		Company *B*	
	Units	Holes	Units	Holes	Units	Holes	Units	Holes
Integrator-summers	16	352	18	432	34	748	30	720
Integrators	14	196	0	0	34	224	0	0
Summers	28	392	27	432	46	644	45	720
Inverters	22	88	15	45	24	96	25	75
Grounded potentiometers	93	186	70	140	165	330	140	280
Free potentiometers	12	36	10	30	20	60	10	30
Relay amplifiers	8	64	6	48	16	128	10	80
Function switches	10	60	12	96	16	96	20	160
Free diodes	16	32	9	18	25	50	15	30
Free passive elements	25	50	12	24	25	50	20	40
Eight-pen recorders	2	16	2	16	3	24	3	24
xy plotters	2	16	2	16	4	32	4	32
Interconnects—trunks—spares	150	150	220	220	190	190	278	278
Multiples	72	72	96	96	151	151	192	192
References—grounds	40	40	84	84	88	88	168	168
Miscellaneous—scope—noise generator	18	18	26	26	23	23	56	56
Transport delay units	2	8	0	0	4	16	0	0
Fixed DFG units	6	12	0	0	12	24	0	0
Electronic multipliers	30	100	12	54	54	180	20	90
Function interpolators	10	20	0	0	18	36	0	0
10-segment DFG	12	72	12	48	24	144	20	80
Servo resolvers	2	52	3	87	4	104	5	145
Servomultipliers	4	88	6	150	6	144	10	250
Total patchholes		2,128		2,078		3,600		3,450

Unshielded cords are generally used with these boards and may cause noise pickup, between-patchcord leakage, and intercircuit capacitances in excess of 10 $\mu\mu$f. Shielded patchcords may be used, but since separate patching of the shield terminations is required, the number of available holes places an upper limit on the number of shielded patchcords that may be used. Molded patchbays are the most economical but do not meet the same specifications as shielded-type patchbays.

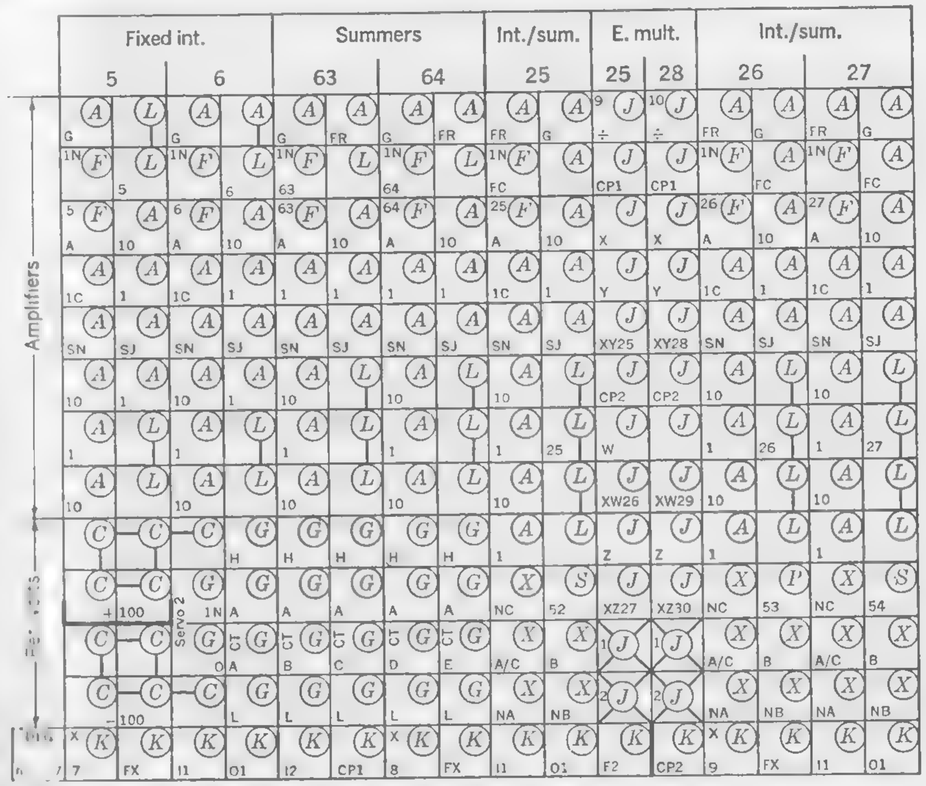


FIG. 4.4.2. Section of a Computer Systems, Inc., CSI 5900 computer problem board. The board layout permits maximum use of single- and double-ended patchplugs to facilitate programming and patchboard switching. *CP* points accept single-ended plugs which ground and thus energize relays to transfer amplifiers associated with multipliers and function generators to operation as inverting amplifiers. *NA*, *NB*, and *NC* accept single plugs to transfer relay control bus to effect separate control of amplifier groups. *SN* points are junctions of gain resistors 10, 1, 10 and are so grouped that they may be connected to the summing junction of the amplifier by a two-point plug or assigned to some other amplifier by means of a patchcord. Amplifier outputs can be patched directly to potentiometers, which may be patched to the input of the following amplifier. A study of the layout will reveal numerous other connections which can be made by patchplugs so as to minimize use of patchcords. Large capital letters denote different colors.

4.4.5. Patchboard Layout and Human Engineering. For maximum efficiency in the use of computer time, the operator must have centralized control of the computer program, mode of operation of switched devices, and input and output monitoring means for static and dynamic checks of computer performance. Setting of coefficient potentiometers produces the greatest time lag between problem insertion and initial problem solution. Larger systems require automatic input-output systems capable of controlling servo-set potentiometers from prepared punched tape and of verifying the settings of the potentiometers against command information (Sec. 4.2).

The greatest contribution to centralization can be made by effective problem-board layout and patchhole utilization. In computer systems whose capacity does not exceed 125 amplifiers, it is preferable to avoid splitting the problem program

between two stations. Insertion of the problem board into the patchbay system must represent a complete address to the computer and the modes of operation of amplifiers, multipliers, dividers, function generators, and other accessory components so that the operator need not have undue concern about remotely located components and the states of associated switching circuitry. Table 4.4.1 gives a general comparison of number of computing devices which may be associated with a patching station while maintaining control of all the modes of their operation. Note that company *A* achieves an increase in amplifier capacity of 33 per cent over that of company *B* although the number of patchholes is approximately equal for their respective program boards. Company *A* employs a patchbay system permitting use of a single-ended ground plug instead of two-hole or four-hole shunt plugs for grounding and control purposes, thereby freeing several hundred holes for use with additional equipment. A further advantage of single-plug patching is that the presence or absence of a plug indicates to the operator that mode of operation of a switched device without the necessity of tracing connections. Figure 4.4.2 depicts a section of company *A*'s problem-board layout which extends the philosophy of plug patching with a minimum utilization of two-ended patchcords.

INTERCONNECTIONS AND GROUND CIRCUITS

4.4.6. Miscellaneous Design Considerations. Machine contributions to the intrinsic errors of the analog computer determine the accuracy of solution and limit the configuration of programming and time scaling that may be used. High performance requires that the dynamic components have high internal gain, wide bandwidth, and good phase margins; such characteristics are more readily achieved in any single component than from an assemblage of components in a computer console (see also Sec. 2.4). Console layout, cabling techniques, and quality of workmanship can become severe limiting factors on computer performance. The most serious causes of inaccuracy are leakage, noise, coupling impedances, and stray capacitances.

4.4.7. Leakage. Analog-computer programming involves interconnection of high-impedance passive elements to high-gain electronic devices which derive their outputs from incremental voltages applied to their input terminals. Leakage resistance paths from other sources to these input terminals must be in the order of 10^{11} ohms if the potential accuracy of the computer is to be achieved. The most rigorous test of computer static accuracy results when the program derives a term of an equation from the difference of quantities whose excursions encompass the full dynamic range of the computer. The importance of leakage contributions becomes obvious when the error in computation is taken as a percentage of a calculated value. Figure 4.4.3 demonstrates that leakage-path contributions can actually become more important than the accuracy of passive elements. While leakage between other connections may be troublesome, the leakage to the amplifier input grid and across component terminations connected to it produces errors of greatest importance. Discussion of patchboard construction indicates means of preventing leakage due directly to patchbay characteristics. Leakage in wiring of the input grid to other components, such as control relays and amplifier connectors, can be obviated by the use of coaxial or shielded wire. Leakage paths of importance can then be fairly well relegated to terminal-to-terminal leakage of components, the degree of importance being determined by the level of potentials existing on neighboring terminals. The most elaborate connectors provide coaxial contacts for amplifier-grid termination and thereby

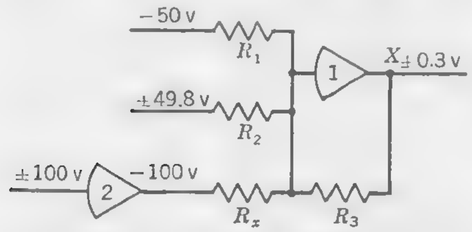


FIG. 4.4.3. If the leakage resistance R_x is 1,000 megohms, the output error is 50 per cent, even though the computing resistors R_1, R_2, and R_3 are ideally matched to 1 megohm. The error in the output X due to leakage exceeds that resulting from a 0.2 per cent deviation in either R_1 or R_2. If R_3 is replaced by a 1-μf capacitor, an integrator drift of 100 mv per sec results.

isolate the grid from other contacts; other connectors separate summing-point terminations from high-voltage leads by grounded contacts; others merely rely on the length of the creepage path across body material with high insulation properties. The last method may be marginally satisfactory if solderless connections are used or if extreme care is taken to avoid flux spillage in soldering connections. The advantages of modern insulating materials such as orlon, nylon, or filled diallyl phtalate are often lost because of surface leakage paths which result from solder-flux residues which are extremely difficult to avoid, particularly on miniature components.

4.4.8. Parasitic Coupling Impedances and Noise. The accuracy of analog-computer solutions is a function of the time scale of the equation program. Precision of passive elements and spurious inputs due to leakage are limiting factors if the program is scaled down in time, and the phase response of dynamic components limits the accuracy if the program is scaled up in time. Neglecting electromechanical components, the time scale that may be used for a given accuracy is determined by parasitic impedances inherent in the computer assembly and its power supplies. The dynamic response of a computer is a direct function of the bandwidth of its amplifiers, which is never less than 10 kc and may exceed 50 kc if high-speed iterative operation is a design objective. The parasitic impedances must then be considered over a frequency range of several megacycles, since they affect the usable bandwidth more than any other factor (Sec. 2.4). The nature of these impedances is complex and their positions in a computing configuration are unpredictable; no reliance can be placed on reduction of their effects by feedback in the closed loops. More often than not, their phase-shift characteristics produce regenerative action and parasitic oscillations and increase the level of high-frequency noise. Parasitic oscillations due to capacitive and resistive coupling between input and output terminals of amplifiers are easily dealt with by proper shielding and component layout. Series coupling impedances that occur in power supplies, wiring, and the computer ground-return system are most troublesome; they are not easily determined quantitatively and can affect both stability and noise level.

The point at which coupling impedances become critical for stability and produce excessive cross talk between amplifiers depends on the current through each coupling impedance. A power supply servicing over fifty general-purpose amplifiers must have an internal impedance in the range of 0.01 to 0.05 ohm over the frequency range of the amplifiers if stability is to be assured for any computing configuration. If redundancies in equipment can be tolerated and multiple power supplies are used to service small numbers of amplifiers, the internal impedances can be somewhat higher. Current distribution and choice of conductor sizes in the power cabling should be determined so that cable impedances are small compared with the internal impedance of the power supplies. Cable layout and lengths should be chosen for minimum self-inductance. Wherever possible, the cable impedance should be enclosed in the voltage-regulator loop so as to divide the effective cable impedance by the gain of the regulator amplifier. The noise level at the regulator output should be less than 1 mv, so that power-supply contributions to amplifier noise may be kept to negligible levels when amplifiers are operating in high-gain loops.

4.4.9. Ground Circuits. The most prevalent cause of parasitic oscillation and high noise level in the computer is coupling in the ground-return system. Since all the potentials in the computer are referenced to the computer ground bus, its impedance embraces both power and signal sources. The distribution of electronic devices in a computer often precludes effective grounding at a single point, so that the ground-system impedance must be low enough to effect an essentially unipotential system. *Such a system can be approximated with copper bussing of sufficient cross section* (1 to ½ sq in.) *to achieve a maximum potential difference of* 50 μv *between any two points in the system under all operating conditions.* For most effective grounding of power and signal terminations, the conductors between the bus and individual components should be of minimum length regardless of cross-sectional area. *Shield grounding should be separately conducted to the bus system,* so that conductor shields do not carry either power or signal currents. *Conductors of current to power devices such as servomechanisms, relays, and stepping devices should be separately conducted to their power*

source and grounded to the system at that point. Separate amplifier-chopper ground connections and separate coefficient-setting potentiometer grounds are highly recommended and may permit ground connections of smaller cross section (No. 10 copper wire). If the necessity of earthing the computer ground system exists, the secondary windings of all transformers connected to the ground system should be electrostatically shielded from the line power source. Utmost precaution in treatment of the ground system is warranted if the computer noise level due to line-frequency pickup, line transients, and computer switching and keying transients is to be kept below 1 mv. High-frequency pulses in the grounding system can be particularly troublesome, since they are generally integrated by chopper-amplifier decoupling networks and produce amplifier offsets depending on pulse height and repetition rate.

4.4.10. Stray Capacitances. Stray capacitances are of importance because of their effect on amplifier frequency response (Sec. 2.4). The design of an operational amplifier is difficult in itself, because neither the magnitude nor the nature of the feedback and input impedances is fixed but may vary widely depending on the requirements of the program. The designer therefore strives in tailoring the response of the amplifier to obtain the widest gain and phase margin. Stray capacitances deteriorate this

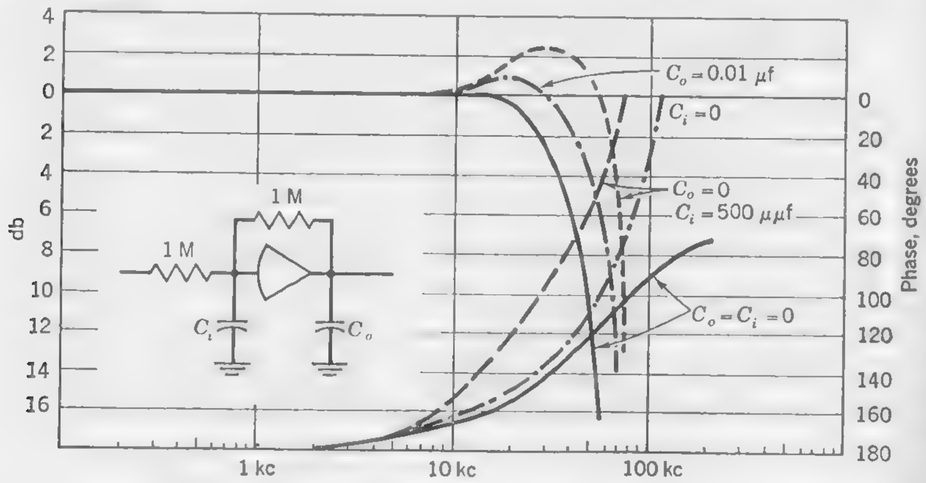


FIG. 4.4.4. Effect of cabling capacitances on amplifier frequency response.

response by either decreasing phase margins to the point where stability becomes marginal or by peaking the amplitude response and thereby limiting the number of amplifiers that may be enclosed in a degenerative loop without parasitic oscillation of the computer. Shunt capacitances are unavoidable, since they are generated by other design considerations. Capacitive loading of an amplifier grid with respect to ground is a function of the shielding required to preclude capacitive coupling and noise pickup. Shielding is accomplished with consideration for wiring lengths and wire specifications, with the percentage shielding of a wire of given dimensions determining the shunt capacity from the grid connection to the outer shield which is grounded. Such capacitance may vary from 11 $\mu\mu$f per ft for coaxial wire to 50 $\mu\mu$f per ft for shielded wire with outer dimensions in the order of 0.090 in. In the writer's opinion, the use of shielded wire should be restricted to wiring between high-impedance passive elements and input terminals, since shielding of low-impedance sources such as amplifier output connections may accomplish little more than increase in capacitive loading. It is important to note that amplifier response must be compensated for capacitive loading of the input grid, and that such compensation imposes the requirement of equal loading of all amplifier input grids if the response of all amplifiers is to be identical. In large computers, where wiring lengths may vary widely, such equalization may be accomplished by selection of wire so that capacitance per foot is proportioned to the length of the wire. Figure 4.4.4 illustrates the effect of wiring capacity on the response of a typical amplifier. Note that, although the response at low frequencies is substantially

unaltered, peaking of response at frequencies that produce considerable phase shift limits the utility of the amplifier to loops that do not enclose more than three to five amplifiers. One can minimize shielding requirements by cable-harnessing practices which separate power conductors from signal wiring and maintain low current densities in cabling, particularly at line frequencies and their harmonics. A-c power wiring to high-voltage supplies, amplifier filaments, and conductors of unregulated voltages to power devices are major sources of difficulty. Power supplies can be located away from computing elements or, preferably, in a separate assembly. Amplifier filament wiring is difficult to deal with, since it is virtually impossible to achieve complete separation of resultant magnetic fields and high-impedance connecting wiring. Multiple filament transformers suitably located help to avoid line-frequency pickup. Extreme precautions in line-frequency wiring is warranted if computer noise levels in the order of 1 mv are to be achieved without excessive shielding, which might deteriorate amplifier response.

COMPUTER ENCLOSURES

4.4.11. Cabinets. The design of analog-computer enclosures involves computer performance, reliability, and human factors.

Two approaches have been taken to the design of computer systems. The concept of a *separate operating console* housing the patchbay system and all scaling, control, and input-output functions aims at separation of the operating functions from those of service and maintenance. Such a design ideally restricts the operator's concern to problem solution and places the maintenance engineer behind the scenes to deal with hardware performance with a minimum of communication between the two functions. While this concept effects a better environment for both the operator and maintenance sections, the physical separation of the equipment from the control console decreases the performance obtainable from the computer because of limitations imposed by series and shunt impedances of extended cables. In such a system the entire computer-room installation attempts to meet computer-design criteria, and economic considerations have limited the number of such installations. The most prevalent design encloses the patchbay system and controls in the same cabinet, with modular electronic units inserted from the operating side of the computer. A compromise between the two concepts is achieved in computers designed so that modular units are inserted from the rear or maintenance side of the machine. Such a compromise effects separation of computing and maintenance functions without physical separation of the equipment. Complete separation of the two functions is obviously not possible, since the computation depends on the operator's cognizance of equipment malfunction. The design should, however, aim at a minimum of interference.

FIG. 4.4.5. In the REAC[R] 400-series computers, electronic components are mounted on U-shaped chasses to allow free flow of air from the bottom to the top of the cabinet, and special ducts channel air from the intake fan in the rear.

Further requirements of mechanical design are similar to those of packaging any electronic equipment of equal complexity. Accessibility of all parts will minimize down time. Modular plug-in units facilitate service through quick replacement of malfunctioning units that may be serviced on auxiliary test jigs. Maintenance of control circuitry is greatly simplified by the use of plug-in relays.

4.4.12. Ventilation. While most electronic devices will perform satisfactorily over a large range of temperatures, the temperature coefficients of passive elements (especially integrating capacitors, Sec. 2.4) require consideration if accuracy is to be

maintained. If sufficient ventilation is not effected, the passive elements must be enclosed in a temperature-controlled oven operating at an elevated temperature. Ovens, however, have considerable thermal inertia and must be energized even when the computer is not in use if temperature stability is to be effected and difficulties due to recycling (Sec. 2.4) and moisture condensation are to be avoided. The necessity for a temperature-controlled environment for resistors can often be avoided by temperature compensation (choice of resistance alloy) in manufacture. Capacitors whose temperature coefficients are in the order of 30 to 50 ppm per μf per deg C are sufficiently temperature-sensitive to warrant concern. Since capacitors are adjusted to close tolerances by mechanical compression, the stability of such adjustment may suffer with temperature cycling in improperly designed ovens. Good ventilation practices can often achieve sufficient component stability, especially since modern computer installations are equipped with air-conditioning systems which can maintain ambient room temperatures within 5°F.

The enclosure-ventilation method illustrated in Fig. 4.4.5 is adaptable to ducting systems. In other designs, a thermal-barrier wall isolates heat-dissipating units from passive elements so as to achieve for the passive-element mounting area a temperature rise in the order of 1°C above ambient room temperature. Air-intake filters are incorporated in most ventilation systems, and the intake fans pressurize the enclosure to avoid infiltration of dust.

PART 5: COMPLETE ELECTRONIC ANALOG COMPUTERS: SMALL AND MEDIUM-SIZED COMPUTERS

By Granino A. Korn

INTRODUCTION

4.5.1. Introductory Remarks. Applications. A general-purpose d-c analog computer will be arbitrarily referred to as *small* if it has less than 20 operational amplifiers. *Medium-sized* d-c analog computers may have between 20 and 60 amplifiers and range from combinations of small computers to installations which differ from really large computers (Sec. 4.6) mainly by the omission of expensive digital programming, readout, and checking equipment (Secs. 4.2 and 4.3).

Users of small and medium-sized electronic analog computers include

1. Industrial laboratories wishing to provide individual engineers with desk-top computers
2. Educational institutions

In industry, small electronic analog computers can give quick answers to many problems requiring only linear computing elements (e.g., control-system design, network problems) and problems requiring perhaps one or two multipliers and/or function generators (e.g., nonlinear oscillations). Even small analog computers can also serve as stand-ins for filters, controllers, or aircraft in various partial-system tests (Sec. 5.6). *Much less well known is the extraordinary power of the small computer in solving a large class of difficult nonlinear problems with the aid of plug-in fabricated diode circuits as the only nonlinear elements.* Such problems can involve limiting, impact, mode switching, sampling, logical decisions, etc. (Secs. 3.3, 5.4, and 6.3).

Educational institutions will use either individual small analog computers or several small patchboards associated with a medium-sized computer for instruction not only in computers and controls but also in many other fields such as physics, differential equations, random processes, and as psychological-test models. A medium-sized installation may give students a greater sense of power and feel for industrial practice, and it is, of course, available for larger research problems. The main advantage of very small computers is their portability.

4.5.2. Design for Expansion. Most computer installations will require *expansion* as users begin to realize the possibilities of the equipment and as new problems arise.

In the specification of a computing system to be designed or purchased, one of the most important initial considerations is an estimate of its largest conceivable expanded size. A d-c analog computer which is not intended to expand beyond the 20-amplifier size can safely employ amplifiers of small current output (5 to 10 ma) and, possibly, a ±50-volt rather than a ±100-volt output range. The cost of such amplifiers is about 70 per cent of that of larger amplifiers suitable for use in expanded installations. Again, smaller computers often justify specification of 1 per cent instead of 0.1 per cent or 0.01 per cent network components. Such savings must be very carefully weighed against the possibility of future expansion, which is, in the writer's opinion, most frequently the more important consideration.

Article 4.5.9 illustrates a practical expansion scheme. A small educational institution can frequently buy successive components of an expandable high-quality computer on successive yearly budgets rather than obtain funds for a complete smaller installation which can be expanded only in a makeshift fashion. On the manufacturer's side, careful design for convenient expansion can open a larger market. Equipment should also be reasonably compatible with analog-computer components of other manufacturers and should, for example, fit standard relay racks and use the standard ±100-volt range.

SYSTEM DESIGN

4.5.3. Amplifiers and Power Supplies. The smallest d-c analog computers may conveniently employ d-c amplifiers requiring two identical high-voltage supplies (±300 volts or ±250 volts). Relatively simple regulated supplies work well. Typical amplifiers are shown in Fig. 6.2.1*a* (octal two-tube plug-in, with a similar chopper-stabilizer module optional) and Fig. 6.2.1*b* (printed-circuit card with stabilizer channel); both amplifiers are suitable for repetitive as well as "slow" d-c analog computers. The same general type of amplifiers and high-voltage supplies are, incidentally, the most suitable vacuum-tube amplifiers for a broad spectrum of control and instrumentation applications of operational amplifiers (Sec. 6.4). Such extra sales can broaden the cost base of a small computer and should be carefully considered in its design; indeed, the small computer may actually be a by-product of a line of operational amplifiers.

In somewhat larger or expandable d-c analog computers, provision of *three* high-voltage power supplies permits more flexible and economical amplifier design (Sec. 2.3), as the amplifier output stage can now supply voltage amplification as well as output current, and requirements on the preceding stage are eased. Most of the dual or quadruple amplifiers designed for larger computers (Figs. 2.5.1 to 2.5.5) are quite suitable for expandable smaller computers (20 to 60 amplifiers). The totem-pole output stages (Sec. 2.3) help to prevent excessive power dissipation in smaller installations. Power-supply voltages of ±300 volts and −450 volts are particularly useful, since they also permit the use of smaller commercially available ±300-volt amplifiers, multipliers, etc.

4.5.4. Nonlinear Computing Elements. Medium-accuracy (0.1 to 0.25 volt) *servomultipliers* and *electronic multipliers* (diode quarter-square, triangle-integration, and time-division types, Sec. 3.2) are commercially available. Servos pose maintenance problems and preclude fast and especially repetitive computation; their single advantage lies in the relatively low cost of producing several products by functions of the servo-input variable. Electronic multipliers are greatly preferable. Chopper-stabilized units are well worth the small extra cost, even though some of the unstabilized commercial units are fairly easy to adjust with their built-in balance meters.

Diode function generators are also available as commercial units with and without their own amplifiers, or the user may set up diode circuits on a patchbay or simulation board. Plug-in bias-setting networks are a relatively economical means of function storage for diode function generators (see also Sec. 3.3).

4.5.5. Patchbays, Simulation Boards, and Control Systems. (*a*) *Committed and Uncommitted (Plug-in) Networks.* In the writer's opinion,

1. General-purpose analog computers comprising 15 or more amplifiers should have

provision for a patchbay arrangement with *committed integrators and summing amplifiers* (i.e., operational amplifiers with computing networks and control circuits built into the patchbay) for speed and convenience in programming, and also to save (relatively expensive) patchbay terminations.

2. Every general-purpose electronic analog computer should permit *uncommitted operation* of at least some amplifiers with plug-in network components and junction diodes; the most convenient method for this purpose involves small extra plug-in patchboards ("simulation boards") with banana-jack terminations. Simulation boards permit very convenient patching of sophisticated *RC* networks and fabricated diode circuits (Sec. 3.3) and are an excellent place to put less frequently used components, such as operational relays, as plug-in assemblies.

Plug-in network components may be 1 per cent types, or they can be adjustable with the aid of an accessory bridge or null voltmeter. Committed patchbays may as well use 0.1 per cent resistors at only slight extra cost.

Uncommitted patching systems should definitely have plug-in resistors rather than built-in resistors with patchbay terminations. The small extra cost of duplicate resistors for a few plug-in patchboards is negligible compared with the almost incredible nuisance of patching, say, a summing integrator with patchcord connections alone, and there is little reason for placing modern low-temperature-coefficient resistors in a patchbay oven. For larger computer setups, one should use committed networks installed behind the patchbay.

(*b*) *Design of Patchbays and Simulation Boards* (see also Sec. 4.4). Even the smallest general-purpose electronic analog computer can have *plug-in patchboards;* the only exception is the case of fast repetitive computers, which require the shortest and most direct wiring possible. While elaborate spring-contact patchbays (Sec. 4.4) are out of the question for small computers, one or two 32-point ribbon-type cable connectors (e.g., Amphenol Blue Ribbon) provide excellent connections for small patchboards (5 to 20 amplifiers). Coaxial connections to the amplifier summing points are collected at an end or at both ends of the ribbon connector and are separated from high-voltage and output connections by ground terminations to prevent leakage. Connectors may be on the cabinet front, or cables can be used between cabinet and patchboards. The connectors can also serve to connect *fixed special-purpose computer setups* which will make a set of general-purpose amplifiers into an inexpensive special-purpose computer (signal generator, servo tester, process controller, test set for psychological or physiological experiments, etc.). A 10-amplifier computer can have connectors for two five-amplifier patchboards.

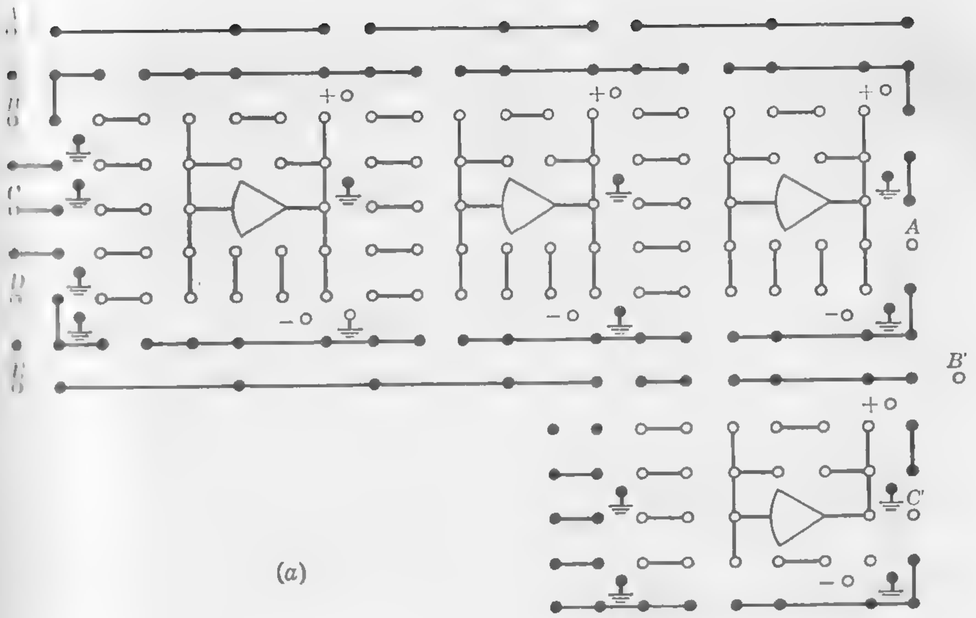
For convenient problem changing, all mode switching of computing elements will be effected through multiple plugs ("bottle plugs") on the patchboard, *not* by switches on the panels of individual computing elements. In this manner, all integrators can be reconnected as summers. Since terminations for integrator control relays add to the cost of the patching system, not all amplifiers on a committed or uncommitted patchboard layout are designed for use as integrators.

In designing patchboard layouts, avoid collecting each type of computing element (summers, multipliers, etc.) together, but *arrange functional groups of computing elements so as to minimize the number of long patchcord connections and to permit shorting-link connections.* A typical functional group might comprise a convenient arrangement of

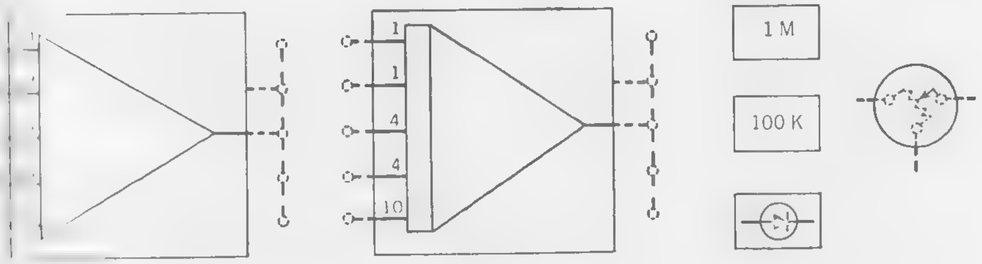
2 integrator/summers	4 to 6 coefficient potentiometers
1 summer	1 to 2 multipliers
1 phase inverter	1 to 2 function generators

to permit solution of a complete second-order equation of motion in a small area of the patchboard. Not all functional groups need contain nonlinear computing elements.

In the writer's opinion, *all patching should proceed from left to right, with return connections through built-in patchbay trunks connecting patchbay terminations at the left and right of the various functional groups.* Patchbays designed in this manner are very substantially easier to work with and greatly help to avoid patching errors.

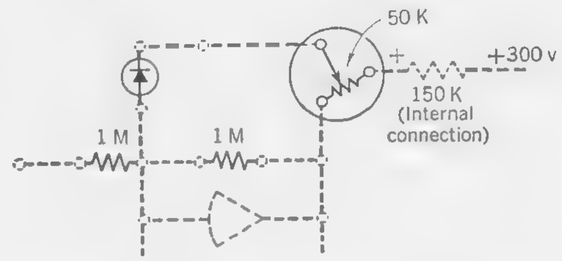


(*a*)



(*b*)

(*c*)



(*d*)

Fig. 4.5.1*a,b*. A simple but versatile simulation board developed at the University of Arizona, Tucson, Ariz. The board connects to the computer by means of a 32-point ribbon-type quick-disconnect connector and admits complete summer, integrator, and relay-comparator plug-ins as well as individual plug-in resistors, capacitors, potentiometers, and junction diodes (Fig. 4.5.1*b*). Most connections can be made with ¾-in. shorting links rather than patchcords. The space shown blank at the lower left in Fig. 4.5.1*a* is for optional terminations of multipliers and function generators, plus a fifth amplifier in an expanded version of the computer.

c,d. Use of three-legged plug-in potentiometers. Terminals marked + and − in Fig. 4.5.1*a* are internally connected through individual 150-kilohm resistors to + and −300 volts, respectively, so that the plug-in potentiometers serve as variable bias supplies for a large number of special diode circuits (*c*) (see also Sec. 3.3) as well as for coefficient setting (*d*). The computer also has panel-mounted 10-turn precision potentiometers, which can be connected, for example between points *A'* and *A* by simple shorting-link [illegible]

Figure 4.5.1 shows a convenient layout for small patchboards, which permits committed operation with plug-in integrator, summer, and comparator modules as well as uncommitted operation. Note how the same type of plug-in potentiometers serve both as coefficient potentiometers and as bias-setting potentiometers for a wide variety of fabricated diode circuits. *Patchboards are preferably made of metal for minimum leakage.* Nylon-insulated press-in or screw-in banana jacks are spaced on ¾-in. centers for use with network components and diodes mounted on General Radio plugs. General Radio shorting plugs replace many patchcord connections.

The same boards can also be used as simulation boards for larger computers.

4.5.6. Control Systems for Small Computers. On the smallest computers, integrator control relays (Sec. 4.1) can be replaced by simple ganged control switches on each patchboard. A slightly more elaborate system uses control relays mounted in the plug-in integrator modules of Fig. 4.5.1*b* and operated from the 300-volt supply through a small push-button control plugged into a telephone jack on each patchboard. If an oscilloscope with a triggered sweep is available, a hand-held COMPUTE push button is the simplest way to obtain repetitive computer operation.

As the computer is expanded (Fig. 4.5.2), these simple controls are easily replaced by a true patchbay-mounted control system. In this connection, it is unwise to save by specifying inexpensive multiple relays for controlling groups of integrators. High-quality reset and hold relays serving at most two integrators will not only permit individual control (Secs. 4.1 and 4.2) but are absolutely necessary for timing accuracy. There is no sense in specifying 0.1 per cent resistors if 0.5 per cent timing errors are permitted.

4.5.7. Metering System and Recorders. Smaller computers commonly utilize *comparison potentiometers* (*null voltmeters*) to set coefficient potentiometers (Sec. 2.1), initial conditions, operational-amplifier transfer functions, and limiter break points; the same comparison potentiometer can also read amplifier output voltages in HOLD. The meter used in the comparison circuit can also function as a center-zero voltmeter for rough voltage readings.

In the smallest machines, the metering circuits connect to a hand-held probe whose cable is plugged into a special patchbay termination. Larger systems may add multipoint or push-button selector switches which select potentiometers, amplifiers, and/or patchbay points for metering. As the system is expanded, provision is made to replace the null voltmeter by a digital voltmeter-ratiometer.

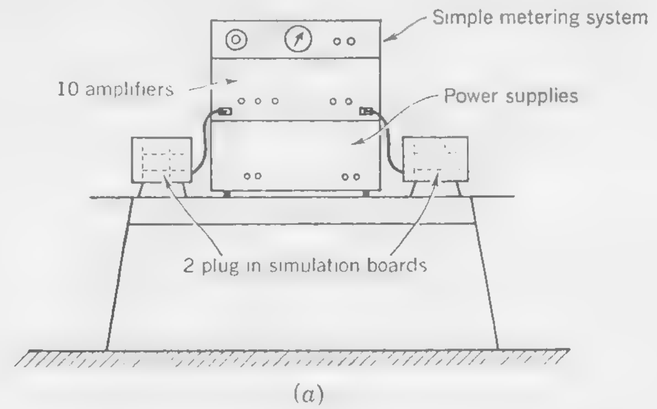
Relatively simple strip-chart recorders and servo plotting boards are available for use with small analog computers and can often be shared with other measuring applications in a laboratory. In the case of the smallest 10-amplifier computers, the cost of the recording instrument may be comparable with the cost of the computer; hence small machines should have provisions for repetitive operation or at least for a push-button compute switch to permit oscilloscope read-out. A Polaroid Land camera permits convenient recording of oscilloscope traces.

4.5.8. Reference Power Supplies. The smallest computers use battery-potentiometer "volt boxes" or small VR-tube stabilized transformer-rectifier packs as isolated sources of constant input voltages, initial-condition voltages, and diode bias voltages. It is preferable to use initial-condition circuits which do not require sources isolated from ground (Sec. 4.1) and to bias all diode circuits in the manner shown in Fig. 4.5.1. One can then obtain ±250 to ±300 volts from the high-voltage supplies and ±105 volts with the aid of VR tubes and set voltages with the aid of a null voltmeter. As the computer expands, an accurate low-impedance ±100-volt reference supply is easily built in the manner of Fig. 2.5.8; note that this circuit operates off the computer high-voltage supplies and does not require a separate transformer-rectifier unit.

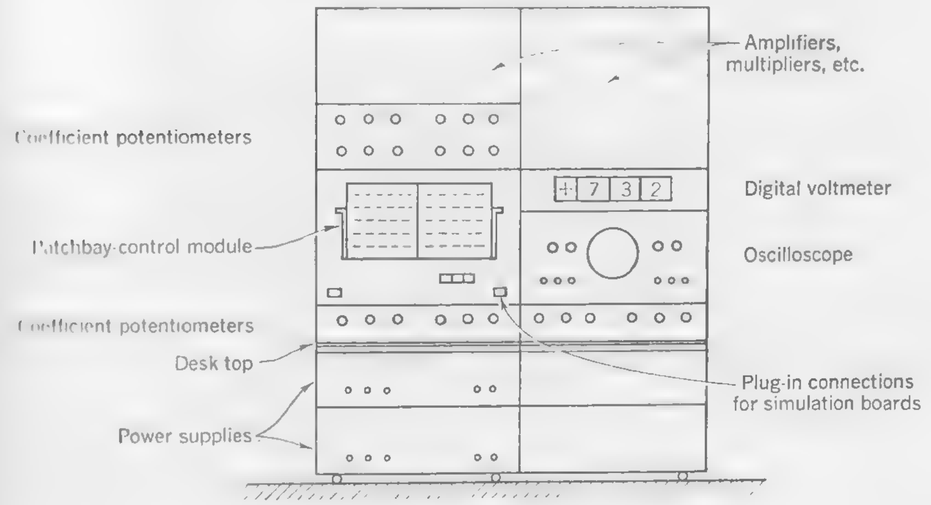
4.5.9. Typical Installations. (*a*) *Desk-top Computers.* A *desk-top computer* useful either for teaching purposes or for use by individual engineers comprises 10 to 20 d-c amplifiers capable of 5- to 15-ma output, a simple metering system, control relays or switches, and interchangeable patchboards for uncommitted amplifiers. Either plug-in or panel-mounted coefficient potentiometers are provided. Such machines are convenient and economical if further expansion is not contemplated; accessory

multipliers and/or diode function generators are mounted in separate desk-top cabinets and can be shared between several desk-top computers as needed.

(*b*) *Expandable Computer* (Fig. 4.5.2). Initial purchase of a 10-*amplifier module* with chopper-stabilized 15- to 20-ma amplifiers together with *power supplies* sufficient for about 20 amplifiers permits desk-top operation with a simple *metering panel*, *patchboards* for 5 or 10 uncommitted amplifiers, and plug-in or panel-mounted *coefficient potentiometers*. *Control switching* for four to six integrators is initially accomplished with simple rotary switches or relays plugged into the patchboards. *Multipliers* and/or *diode function generators* are next added as needed.



(*a*)



(*b*)

FIG. 4.5.2. Ten-amplifier table-top computer (*a*) and expanded installation (*b*).

The next major purchase is a *patchbay-control module* which admits *committed* computing networks for up to 30 amplifiers and controls with high-quality integrator relays for 12 to 16 integrators; a patchbay oven is optional. One may now add a *digital voltmeter* (this is invaluable for setting coefficients and initial conditions) and *additional amplifiers, multipliers, function generators, and power supplies* as funds become available; the 30-amplifier patchbay permits utilization of up to 40 amplifiers if the uncommitted patchboards are used as "simulation boards" for special networks, diode circuits, etc. It is of interest to note that two systems of this type are easily slaved together (Sec. 4.1) into a very respectable 70- to 80-amplifier computer.

PART 6: COMPLETE ELECTRONIC ANALOG COMPUTERS: LARGE INSTALLATIONS

By STANLEY ROGERS

INTRODUCTION

4.6.1. Large Analog Computers. Because of the all-parallel operation of analog computers, solution of large problems (see also Sec. 5.5) will not increase the solution time but requires larger computer installations, which differ qualitatively from smaller ones. Differences arise from their high cost and large size, the difficulties involved in being sure the computer is giving correct solutions to large problems, the various fields in which the computers may be used, and the problem of staffing and administering a large analog-computer installation.

MAJOR DESIGN OBJECTIVES

4.6.2. Requirements. A large analog computer should:
Be big enough to solve the largest problem anticipated
Be big enough to handle anticipated volume of problems
Permit simultaneous solution of several unrelated problems of different sizes at the same time
Make efficient use of all computing elements
Have sufficiently high component accuracy to provide solutions of acceptable accuracy in the usual applications
Be reliable in operation and easy to check out

4.6.3. Simultaneous Solution of Different Problems. The computer must be divided into two or more "stations" or "bays," each comprising 20 to 60 amplifiers and other computing elements, a patchboard, control system, and recorder. A larger number of smaller stations provides greater flexibility in handling problems of different sizes; the control system should provide for slaving any of the stations to any one station. Disadvantages are increased cost and the nuisance of having larger problems divided among many patchboards and stations. For the smaller "large" computers (100 to 200 amplifiers) two to six stations are usually satisfactory. Probably the number of amplifiers per station should not exceed 60, even for the largest computers (600 or more amplifiers).

4.6.4. Efficient Use of Computing Elements. If all stations are alike, assignment of problems and maintenance procedures are simplified, but more efficient use of the installation suggests at least two types of stations. At Convair-Astronautics, for example, the main facility consists of three identical groups of five stations each. Four of each five are so-called "linear" stations (34 amplifiers) and the fifth is a "central" station. It has most of the multipliers (82 products), function generators (21), etc., of the whole group of five, and also has a smaller number (28) of amplifiers, which are usually needed in conjunction with these nonlinear computing elements. The central station has 600 trunks connecting it to each of its four "linear" stations and to the other "central" stations.

This organization of the computer has proved to be very satisfactory for aircraft and missile simulation. The "linear" stations are adequate for many problems. Problems requiring much nonlinear equipment have ready access to it at the "central" station. Very large or very nonlinear problems may require two or all three of the central stations plus a number of the linear stations. Each central station can parcel out its nonlinear equipment in any desired manner through suitable patchbay connections.

The Convair–San Diego analog computer uses an entirely different approach. The control busses (HOLD, RESET, etc.) and all control relays associated with integrators, multipliers, resolvers, etc., are brought out to patchbay terminations. An operator

can thus use trunk interconnections to "borrow," say, an integrator from another station and control it. Such an arrangement can utilize otherwise idle equipment, but makes the problem solution depend on a patchboard wired and in place at another station.

4.6.5. Accuracy Requirements for Large Problems (see also Sec. 5.5). For comparable solution accuracies, the resistors and capacitors in gain-determining circuits of large computers should be more precise than those in small computers. The extra cost of *temperature-controlled ovens* for input and feedback resistors and capacitors is usually justified for computers intended to solve large problems or to have maximum accuracy. A large computer should include means to check positive and negative *reference voltages* at each station against each other and against those in other stations.

4.6.6. Reliability and Checking Features. Reliability in a computer demands that components be of high quality and be conservatively operated. Designs striving for the utmost in performance are not likely to give trouble-free service; the same applies to new designs which have not been proved by a substantial period of service. Since manufacturers vary considerably in the effectiveness of their quality-control and inspection procedures, it will pay to consult with other computer users before buying large amounts of equipment from a new supplier. Proved *check-out and maintenance* procedures for large analog computers are described in Sec. 4.7; *problem checks* are discussed in Sec. 4.3.

To ensure maintainability and simple check-out procedures and to minimize faults and errors, a large analog-computer installation should:
Be built with conservatively designed elements
Contain a number of each type of power supply
Use a minimum number of different types of power supplies, amplifiers, etc.
Have well-regulated transient-free electrical power
Provide special test and check-out patchboards (Sec. 4.7) and special bench-test equipment (Sec. 4.7).

PROBLEM SETUP AND STORAGE

4.6.7. Requirements. A large analog computer should:
Provide means for setting up problems and checking them away from the computer
Have permanently wired subroutines
Have automatic or semiautomatic means for setting potentiometers
Have digital voltmeter and printer for accurate readout and for "dumping" problems at end of day
Provide means for removing problems from the computer, storing them, and returning them to the computer quickly and easily
To ensure convenient operation, the machine should:
Include in strip-chart recorders means for automatically recording zero, sensitivity, paper speed, and time markers (Sec. 4.2)
Start problem solutions in coincidence with a timing mark
Have, for some applications, means for automatically programming runs and changing a few parameters (Sec. 4.2)
Have illuminated push buttons for controls; indicator lights should indicate completion of the control function, and not merely that the button was pushed
Patchboards should:
Contain the whole problem (few or no initial-condition and function switches should be on the control console)
Provide for unclutterd wiring (maximum use of bottle plugs)
Minimize intercircuit leakage

PERSONNEL AND ADMINISTRATION. TYPICAL INSTALLATIONS

4.6.8. Operators. Under the best of circumstances, analog-computer operators would:

Understand the physics of problem
Have feeling for the mathematics of problem
Know analog methodology
Know capabilities and shortcomings of the computer
Use equations suited to getting answers desired
Make good compromises in setting up problem
Have patience to check problem
Develop problem in steps
Have skill in debugging
Be carefully selected and be trained on the job (the best operators are not necessarily the most highly educated)

4.6.9. Administration of Large Computer Facilities. Persistent disputes about administering large analog computers concern *open-shop operation* ("customer" works with computer) vs. *closed-shop operation* (computer personnel run computations), as illustrated by Table 4.6.1. The *co-op shop* teams the man who knows the problem with a professional operator who knows the machine and computer methodology. This combination, if practicable, is more effective than either open or closed shop.

Table 4.6.1. Open-shop vs. Closed-shop Operation of Analog-computer Installations

Open Shop—Advantages	*Closed Shop—Advantages*
Operator understands problem	Operator understands how to get most out of machine
Operator understands significance of solutions	Operator turns out many solutions
Operator avoids making useless runs	Operator knows analog techniques, is skilled programmer
Operator gains insight into problem	Machine does not stand idle while operator ponders over a solution
Operator can best judge effect of compromises in problem setup	Machine can be operated efficiently all day long
Machine and operator form problem-solving team	
Operator is likely to observe effects of a machine fault quickly	

Since the computer staff is usually not in a technical or administrative position to judge whether any given problem should go on the machine, other means for controlling *use of the computer* are needed. If the computer is available without charge (i.e., facility costs are charged to overhead), unimportant problems may saturate the computer's time. If facility costs are charged direct to problems on the computer, many problems that could be solved may not be, even though the machine stands idle a substantial part of the time. One possible solution: reduced charges for using the computer during off hours. Direct charge for computer time is necessarily a far more effective control than the mere requirement of supervisory approval from the department originating the problem.

Where analog-computer time is a direct charge against problems solved, charges for maintenance labor, electric power, depreciation (of company-owned computers), etc., are often collected in a distributive account and assessed on a monthly or annual basis against the appropriate accounts. The annual basis is more equitable, since heavy parts purchases in one month may result in disproportionate charges against certain problems. Both systems are in use.

A large computer facility should have a permanent maintenance staff (Sec. 4.7). Depending on the size of the computer, the number of shifts it operates, the accuracy and reliability demanded by operators, the effectiveness of temperature, humidity, and dust control, and the age and condition of the computer, maintenance costs vary widely. Where the highest standards of performance are demanded on multishift operation, annual maintenance costs may run from 15 to 20 per cent of the capital cost of the computer.

Since the computer art is a rapidly developing one, existing machines may be regarded as obsolete or obsolescent. Some companies owning large analog computers

Table 4.6.2 Comparative Size of Various Types of Large Analog Computers
Averages in Each Installation

Equipment	Berkeley Computation Center, Los Angeles	Convair, San Diego	Convair, Pomona	Convair Astronautics	Dian Labs, New York	Electronic Associates, Inc., Computation Center, Los Angeles	Electronic Associates, Inc., Computation Center, Princeton	The Martin Company, Orlando (Goodyear)	Humble Oil, Baytown, Tex. (Computer Systems, Inc.)	Lockheed Aircraft Corporation, Sunnyvale	Project Cyclone, Reeves Instrument Co.	Space Technology Labs, Los Angeles (EAI)	Sylvania Mountain View (Goodyear)	W.A.D.D. (Reeves) Wright-Patterson Air Force Base	Averages
No. of operational amplifiers	206	560	372	556	444	228	220	428	216	938	664	348	150	488	149
Potentiometers	146	210	180	151	184	167	109	166	148	110	144	126	108	141	8.71
Servo multipliers (shafts)	0	11.4	12.1	6.47	12.2	14.5	13.6	0.935	1.85	4.05	7.68	14.9	0	19.7	3.56
Servo resolvers (shafts)	1.46	2.86	4.03	4.32	2.03	3.95	5.45	3.50	0	2.77	6.48	4.02	3.33	3.28	
Electronic multipliers (total products)	28.2	12.0	17.7	8.63	2.25	8.77	13.6	33.4	27.8	28.8	0.602	7.18	24.0	4.92	14.8
Function generators $y = f(x)$	10.7	4.29	14.5	11.5	3.60	5.26	9.09	7.01	9.26	7.78	3.01	7.18	2.67	1.64	6.74
Function generators $z = f(x,y)$	0	12.5	5.91	*	No data	2.19	0.455	1.17	2.78	0.107	0	0.287	0	1.64	2.47
Limiters (all types)	0	25.7	17.2	34.5	25.2	5.26	8.18	0.701	0	12.4	16.9	4.60	0	4.92	14.0
Diodes (free)	3.88	34.3	45.7	21.6	Many	Many	Many	Many	23.2	8.00	Many	Many	13.3	8.20	19.4
Comparators (relay amplifiers)	5.83	17.1	21.0	17.3	14.0	8.77	11.8	2.34	14.8	8.00	9.04	16.1	No data	6.56	11.6
Strip-chart recorder channels	5.83	15.7	14.5	28.8	5.41	11.4	14.6	8.41	5.56	12.2	7.83	8.62	8.00	7.38	11.8
Function switches or relays	3.39	2.86	12.1	55.4	Many	26.3	11.8	4.67	14.8	25.6	5.12	6.03	8.00	6.56	15.9
x-y plotters	0.971	0.357	0.806	1.62	1.13	2.19	2.73	0.701	1.85	1.39	1.51	2.01	2.67	2.46	1.46
Decade boxes (resistance and capacitance)	0	17.1	25.8	37.2	No data	0	0	2.34	0	3.41	1.51	2.30	0	16.4	10.0
Noise generators	0.485	0.357	0.806	0.360	0.225	1.32	1.36	0.467	0.463	0.853	0.452	1.15	1.33	0.410	0.63
Digital voltmeters	0.971	0.357	1.61	1.98	0.450	1.75	1.36	1.17	0.926	1.39	0.602	3.16	0.667	1.02	1.22
Servo-set potentiometers	No	Yes	Yes	No	No	No	No	Yes	Yes	Yes	Yes	No	No	Yes	
Automatic input/output system	Yes	†	Yes	†	No	No	No	Yes	Yes	Yes	Yes	No	No	Yes	
Transport delay	1.46	No	No	*	No	No	No	No	No	No	0.151	No	No	No	

Note: Several installations make multiple uses of various equipments (such as multiplying by means of tapped potentiometers on servomultipliers, generating functions of two variables by means of two or more one-variable function generators). As far as possible, such duplications have been omitted from the table, the equipment being listed once only under its principal function.

* Under development at installation.

† "Partial."

find that it pays to have engineers on the maintenance staff to keep the computer up-dated and to redesign troublesome circuits.

A computer installation *mixing several makes of computing equipment in one facility* will be more versatile and accurate but more difficult to operate and maintain, and some equipment may have to be modified.

4.6.10. Statistics on Selected Computer Installations. Table 4.6.2 indicates equipment distributions in typical large analog facilities. Quantities of various computing elements are given in terms of the number per 100 amplifiers in order to make comparisons easy. The sizes of the various installations are indicated by the number of amplifiers (first line of the table).

As far as possible, the number of amplifiers listed in the table is the number available to the operator for use for summing, inverting, and integrating. Those amplifiers which are permanently wired to function generators, multipliers, etc., have been excluded when the information supplied for the table made this possible.

PART 7: MAINTENANCE PROCEDURES AND TEST CIRCUITS

By HAROLD L. EHLERS *and* WALTER HOCHWALD

INTRODUCTION

4.7.1. Planned Maintenance. Prime objectives of a planned computer-maintenance program are (1) to maximize the total productive time, i.e., to minimize the computer time which is required to perform periodic maintenance checks and repair; (2) to minimize failures during operational periods, i.e., to maximize the mean error-free operating time of the computer.

As an aid in achieving the first objective, modern computer design employs readily removable (plug-in-type) components. With adequate spares and appropriate off-line test equipment, the major maintenance effort may be performed without penalty to computer operational time. As a further aid, many computing facilities use specially constructed test boards which permit the maintenance technician to check the functional condition of the computer.

To achieve the second objective, a planned preventive-maintenance program must be provided. Predicting a component failure requires a more subtle type of test than that required to determine when a failure has occurred. Consequently, methods proposed in this area are more subject to controversy. One approach used for preventive maintenance consists of applying marginal tests of a component in place. A second method consists of removing the component from the computer at fixed periods to perform comprehensive tests by means of the off-line test equipment.

TEST CIRCUITS

4.7.2. Operational Amplifiers. Operational amplifiers must have high open-loop gain and low noise level, input current, and drift. The following tests compare measurements of these quantities with established standards.

(*a*) *Open-loop Gain.* It is difficult to measure the high open-loop gain of most modern operational amplifiers (generally between 10^6 and 10^9) directly. An approach to measuring the open-loop gain is to measure the gain of stabilizer channel and d-c amplifier independently.

The gain of the chopper or stabilizer amplifier may be measured using the normal amplifier balance controls to:

1. Disconnect the stabilizer-amplifier output from the d-c amplifier
2. Introduce a small unbalance to the d-c amplifier
3. Reconnect the stabilizer-amplifier output
4. Measure the new unbalance

The gain of the stabilizer amplifier is the ratio of the initial unbalance to the stabilized unbalance.

The gain of the direct-coupled amplifier (generally 10^4 to 10^6) is also difficult to measure directly. A more satisfactory approach is to measure the closed-loop dynamic response and, from the measurement, to calculate the open-loop gain. Because the closed-loop phase lag is directly dependent on the open-loop gain, a short-cut method is to check the closed-loop phase lag at a single representative frequency above the stabilizer-amplifier cutoff frequency. A typical frequency for a high-performance amplifier would be 100 cps. The measured phase lag is compared with a preestablished standard to determine if the direct-coupled amplifier gain is acceptable. The phase lag may be measured by a commercially available phase meter or by a bridge circuit[1] (Fig. 4.7.1).

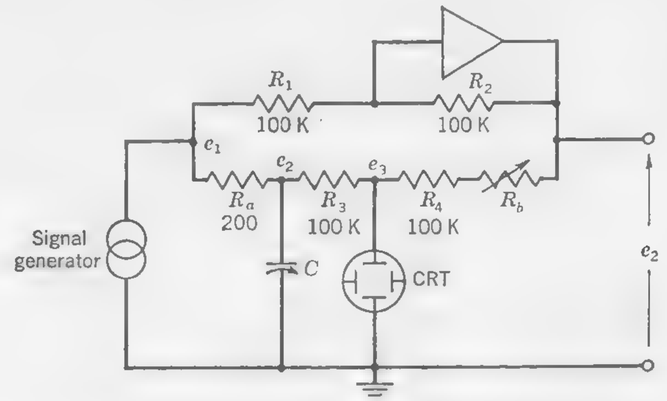


FIG. 4.7.1. Test for amplifier phase lag.

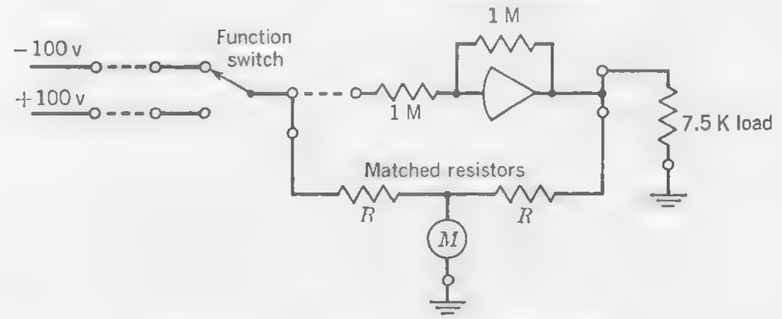


FIG. 4.7.2. Test for amplifier static gain.

In Fig. 4.7.1, one matches R_1 and R_2 (as well as R_3 and R_4) for resistance and capacity by building up the capacity of one of these with conducting ink until the two resistors can be interchanged without noticeable effect on the bridge balance.

A 100-cps signal is applied at e_1. The variable capacitance C and resistance R_b are adjusted for a null at e_3 on the cathode-ray tube. The phase shift between e_1 and e_2 is related to the value of the variable capacitance at null. A maximum acceptable value of C is preestablished for correspondence to the closed-loop phase shift, and thus to the open-loop gain desired.

(*b*) *Closed-loop Gain.* If the dynamic test of the amplifier open-loop gain, as described above, is inconvenient to perform, a simpler but less thorough check can be made of the closed-loop static gain at rated load. This test primarily checks the condition of the output tubes. Because of the inherent gain stability of the feedback amplifier, this test does not adequately check the amplifier open-loop gain. This test is not suitable for preventive maintenance, because it does not check for marginal vacuum-tube performance. The test may be made by using matched resistors and a null meter (Fig. 4.7.2). Alternatively, the amplifier output may be read directly by means of a digital voltmeter.

(*c*) *Noise.* Amplifier noise is best measured by observation on a cathode-ray-tube oscilloscope. The low-frequency noise is primarily due to harmonics of the a-c supply coming into the amplifier through stray couplings. The high-frequency noise (above 1 kc) consists of random tube noise due to microphonics and resistor noise. Excessive high-frequency noise is usually caused by a faulty tube.

(*d*) *Input Current and Drift.* If the amplifier is balanced by using the available balance controls with the amplifier wired as shown in Fig. 4.7.3, then the offset voltage as measured at the output is proportional to the grid current. (Note that a failure of the stabilizer amplifier could also result in an amplifier offset voltage.) If the stabilizer amplifier is operating properly and the unbalance does not exceed that which can be compensated for by the stabilizer amplifier, the amplifier drift should be within the design specifications. A thorough test of amplifier drift can be made only by observation of the amplifier unbalance over a substantial period of time.

(*e*) *High-gain Test.* A simple test which reveals marginal performance of an amplifier is to measure the noise and balance of an amplifier with reduced feedback gain by using the test setup of Fig. 4.7.4. Under these conditions the closed-loop amplifier performance is more sensitive to a loss in open-loop gain. This is indicated by the inability of the amplifier to balance properly and/or excessive noise level at the amplifier output.

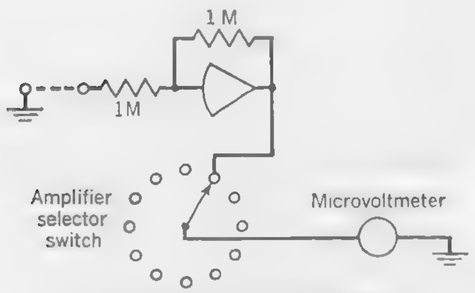


FIG. 4.7.3. Test for amplifier offset.

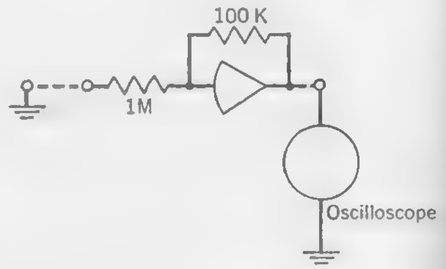


FIG. 4.7.4. Test of amplifier with high feedback gain.

4.7.3. Reference-voltage Supplies. Before initiating a test of the passive elements, the computer reference-voltage supplies should be accurately set, say by means of a standard cell, a precision potentiometer (such as the Leeds & Northrup K-3 potentiometer), and a null meter; this method permits an over-all accuracy of 0.01 per cent or better in setting the reference voltages.

A digital voltmeter comprising its own standard cell for calibration purposes may also be used for setting the reference voltages.

If the plus and minus reference-voltage supplies are accurately balanced against each other, an absolute error larger than 0.01 per cent may be tolerated without corresponding degradation in computing precision; Fig. 4.7.5 shows a practical balancing circuit.

The switch is successively placed in the up and down position, and either the plus or the minus reference supply is adjusted so that the output voltage with the switch in the up position is the negative of the output voltage with the switch in the down position. (Note that the two summing resistors on the first amplifier need not be matched.) An 0.01 per cent unbalance between the ± 100 reference supplies results in a 1-volt unbalance at the output.

4.7.4. Passive Elements and Control Relays. The accuracy of an operational amplifier depends critically on the precision of the resistors and capacitors used to perform the summing and integrating operations. It is equally important that the computer control relays function properly. Modern analog computers have specified accuracies for the operational amplifiers as high as 0.01 per cent. To check the passive elements to this precision requires the use of a special test setup and is time-consuming. Consequently, in most computing facilities, precision testing is performed only

once or twice a year. Tests for gross types of errors (resistor shorts, open circuits, and the like) may be performed daily.

(*a*) *Matched Resistors.* The test setup of Fig. 4.7.5 may also be used for matching resistors. If the output voltage at e_o is less than 0.01 volt, then input resistors are matched to within 0.001.

(*b*) *Precision Resistor Testing.* For precision testing of resistors, a bridge employing standard and matched resistors and a null meter is used (Fig. 4.7.6). The null meter

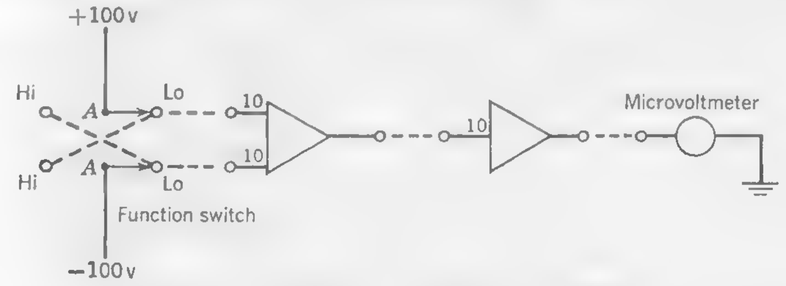


FIG. 4.7.5. Alignment of reference supplies.

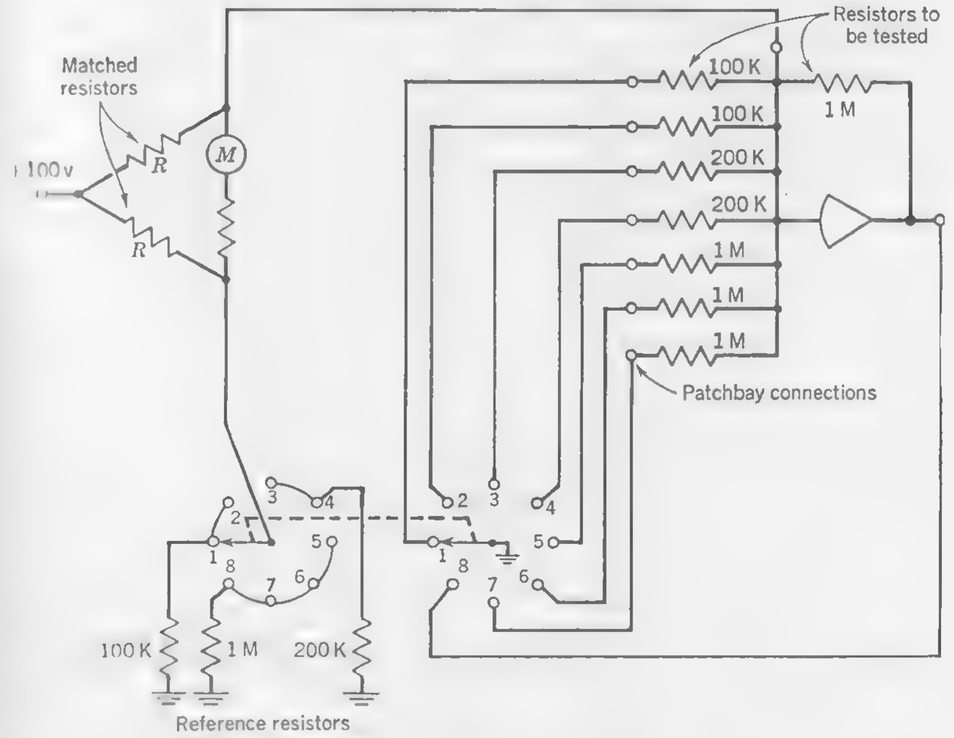


FIG. 4.7.6. Precision test of summing resistors.

should be calibrated so that a given meter reading corresponds to a known error. Alternate means for performing the same calibration with the aid of operational amplifiers are shown in Fig. 4.7.7. An 0.01 per cent error in the test resistor results in a 1.0 volt error at the output.

(*c*) *Testing for Gross Resistor Errors.* Tests for errors in the summing resistors of the order of 0.5 per cent or greater can be performed more rapidly by tests designed to check several input resistors simultaneously.

One test circuit (Fig. 4.78) applies plus and minus source voltages to equal input resistors. If the voltage error at the amplifier output exceeds 100 mv, the resistors

are individually tested. A 100-mv error could be caused by an error of 0.01 per cent in a 100-kilohm resistor, or 0.1 per cent in a 1-megohm resistor.

Figure 4.7.9 illustrates a second test circuit designed to perform a similar function. If the output as read on a digital voltmeter is in error by 20 mv or more, the resistors are individually tested. It can be shown that a 20-mv error could be caused by a

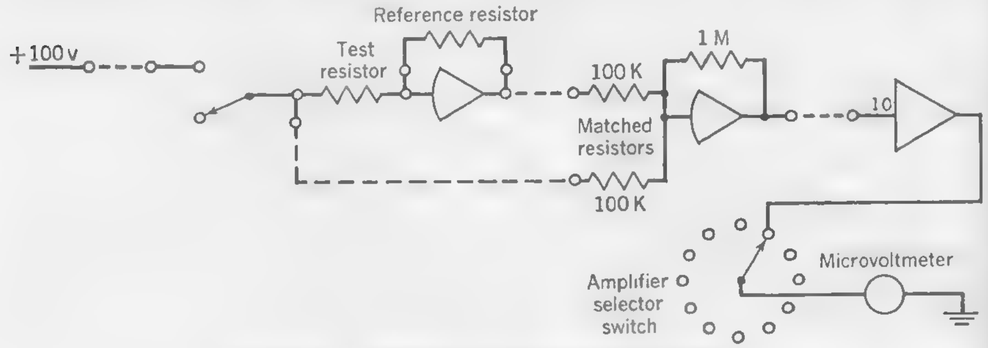


FIG. 4.7.7. Precision test of summing resistors.

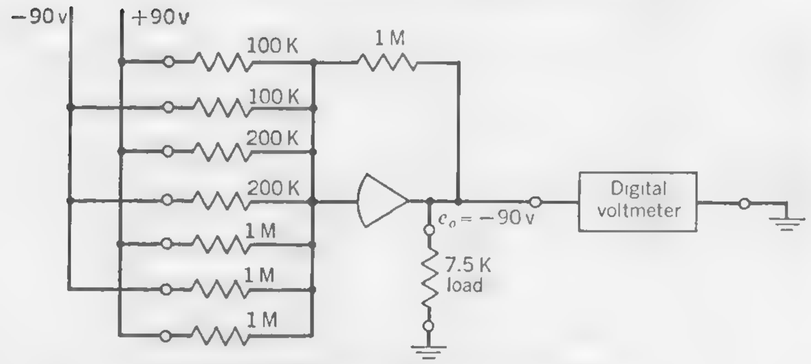


FIG. 4.7.8. Gross test of summing resistors.

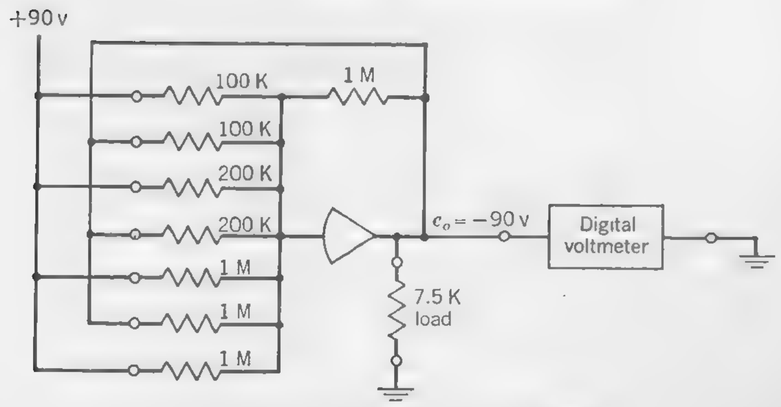


FIG. 4.7.9. Gross test of summing resistors.

0.015 per cent error in a 100-kilohm resistor or a 0.4 per cent error in 1-megohm resistor. The circuit shown has the weakness of employing a low feedback resistance or high feedback gain; some amplifiers would oscillate with the low feedback impedance shown. If the amplifiers oscillated, resistors would successively be transferred from the feedback to the input path until the amplifier stabilized. This transfer would be done at the penalty of increased gain, reduced input voltage, and consequently

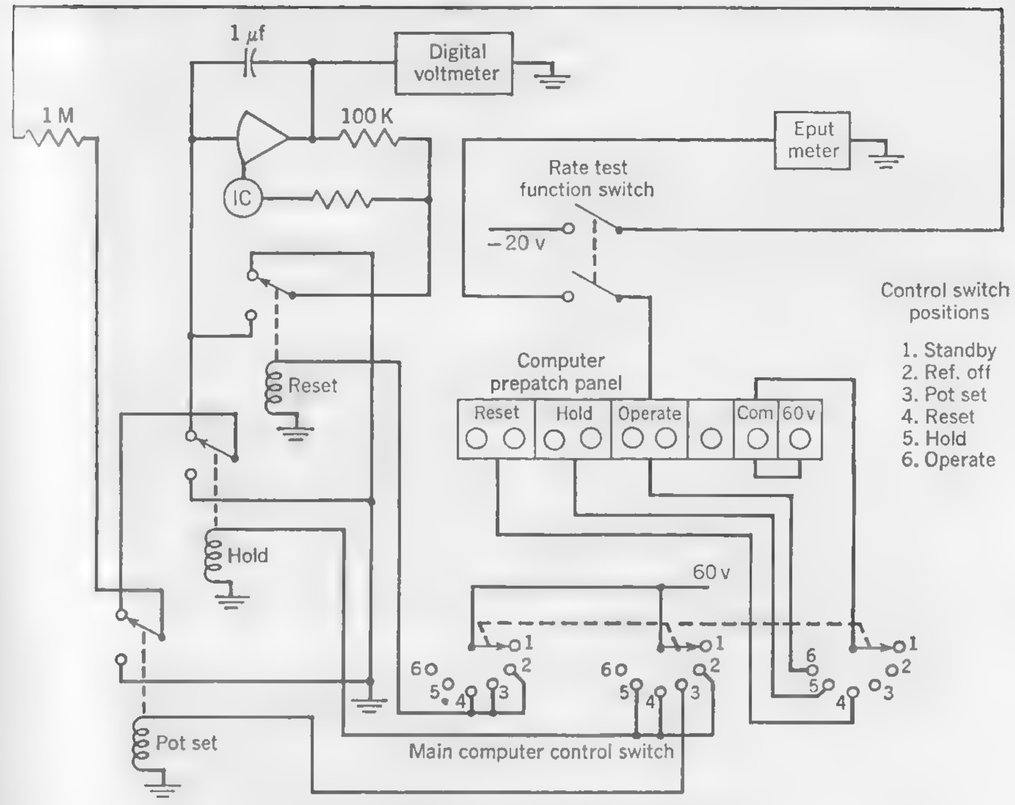
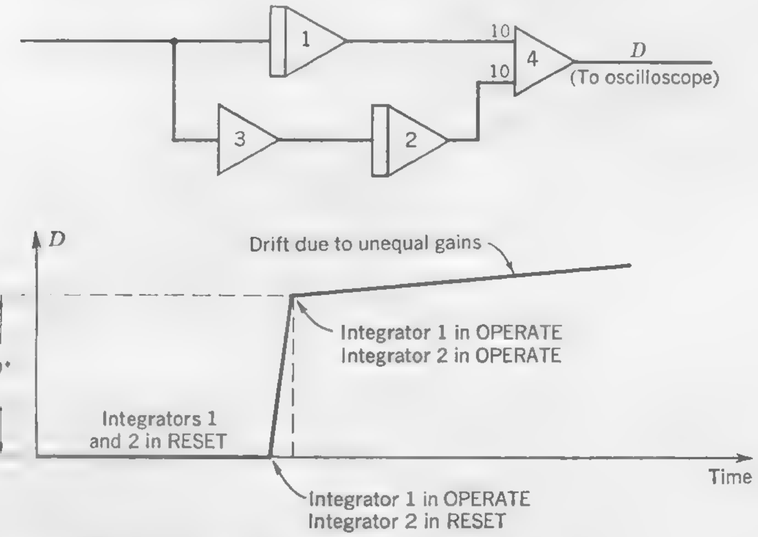


FIG. 4.7.10*a*. Test for amplifier integration rate.



D^* = Voltage due to difference in timing of relays in integrators 1 and 2.
1 volt corresponds to 1 millisecond.

FIG. 4.7.10*b*. Relay-timing test. (*Contributed by B. Loveman, Reeves Instrument Corp., Garden City, N.Y.*)

decreased resolution for the test. Note that the amplifier stability, under the conditions of the above test, may be dependent on the sign of the input voltage.

The test procedures described can be executed rapidly if prewired test boards are used.

(*d*) *Passive Elements and Computer Control Relays.* For precise computation all integrators must start simultaneously and must integrate at the same rate. Hence not only the feedback capacitors and input resistors, but also the make-to-break times of the OPERATE-RESET and HOLD relays must match.

Test circuits for checking the integrator rate and the action of the RESET and HOLD relays are shown in Fig. 4.7.10. In Fig. 4.7.10*a*, a prepatch board is wired so that all

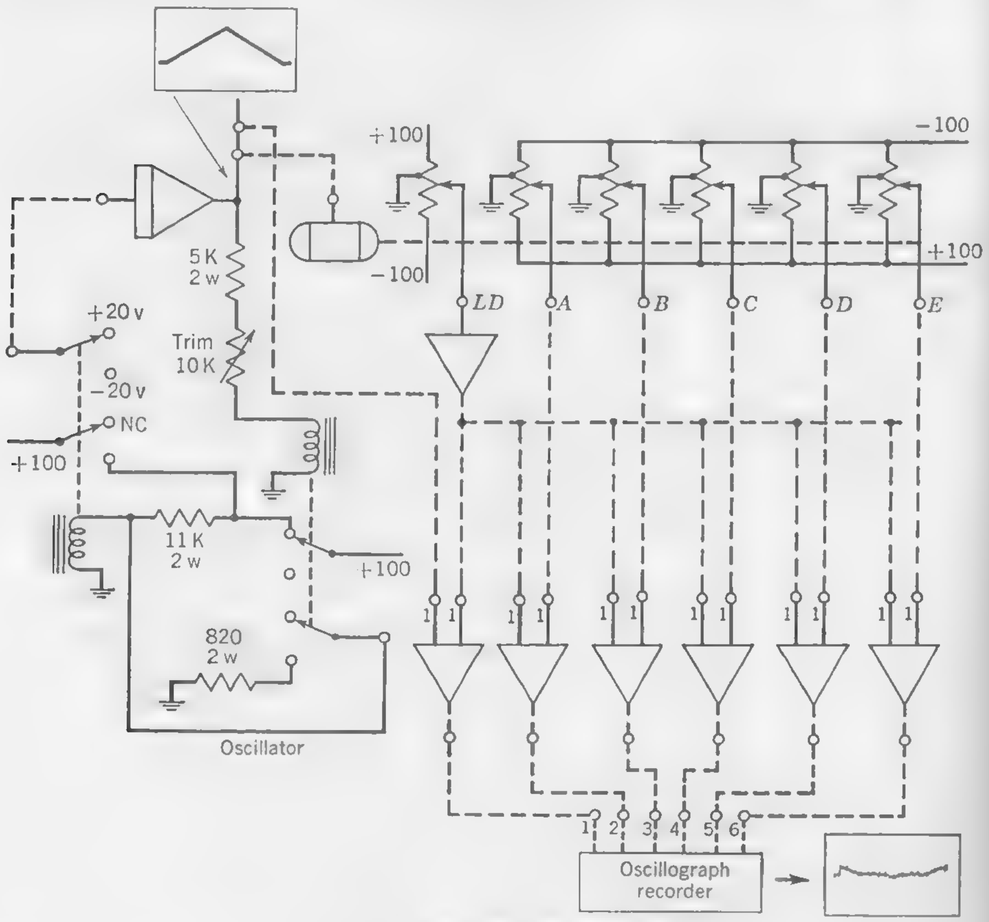


FIG. 4.7.11. Test of servomultiplier.

integrator amplifiers are connected as illustrated. With the function switch in the UP position, the computer control switch is turned from RESET to OPERATE. When the amplifier has integrated to approximately +90 volts, the computer control switch is placed in the HOLD mode. A comparison of output voltage of all integrating amplifiers indicates the match of the amplifier integrator rates. A variation in amplifiers of 20 mv indicates a 0.01 per cent difference in the effective integration rate. An error of this magnitude could be caused by a 0.01 per cent error in feedback capacitance, or by a 1-msec difference in relay throw time.

An absolute check of integrator rate can also be made with an eput (events-per-unit-time) meter. The count would be initiated by the voltage applied when the control switch is placed in the OPERATE position and stopped when this voltage is removed in the HOLD position. With eput meters capable of time measurement to 1 msec, the

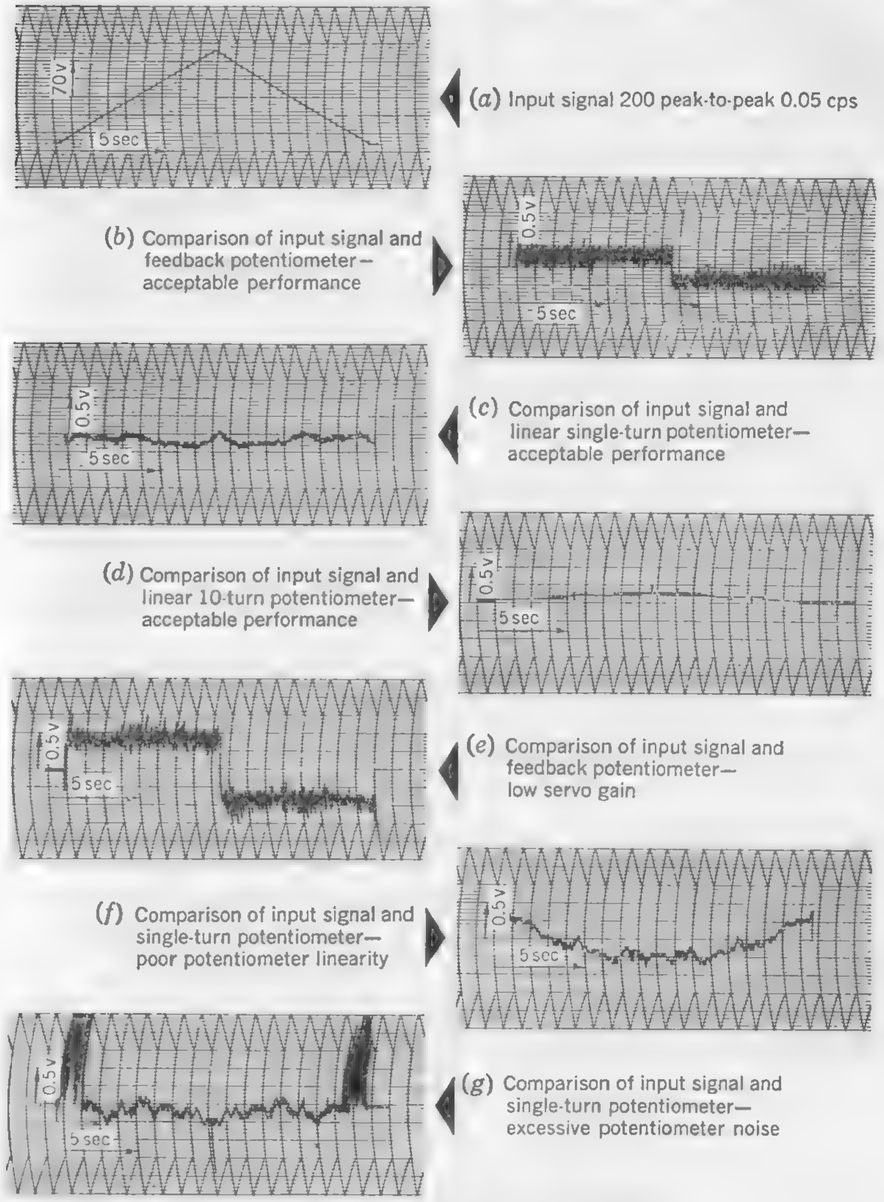

70 v
5 sec
(a) Input signal 200 peak-to-peak 0.05 cps
(b) Comparison of input signal and feedback potentiometer—acceptable performance
0.5 v
5 sec
(c) Comparison of input signal and linear single-turn potentiometer—acceptable performance
0.5 v
5 sec
(d) Comparison of input signal and linear 10-turn potentiometer—acceptable performance
0.5 v
5 sec
(e) Comparison of input signal and feedback potentiometer—low servo gain
(f) Comparison of input signal and single-turn potentiometer—poor potentiometer linearity
0.5 v
5 sec
(g) Comparison of input signal and single-turn potentiometer—excessive potentiometer noise
0.5 v
5 sec

FIG. 4.7.12. Test records of servomultiplier.

largest error would be due to the resolution in measuring the 20-volt input. Assuming measuring instruments with a precision of 10 mv, this would result in a 0.05 per cent error in the measurement of integrating rate.

The error caused by differences in the make-to-break time of the relays may be removed by placing the computer switch in the OPERATE position and by using the function switch to apply the input voltage to all integrators simultaneously. In a similar way the input voltage may be removed simultaneously from all amplifiers. Thus the function switch can be used to isolate the capacitor-resistor errors from the relay errors.

A circuit that can be used for comparing the relay timing and integration gain of two amplifiers is shown in Fig. 4.7.10*b*. The circuit sensitivity is such that a 1-msec error in relay timing would cause a 1-volt offset between integrators.

The make-to-break time of the potentiometer-set relay is not as significant as that of the RESET and HOLD relays. The potentiometer-set relay can be checked by putting 100 volts into a unity-gain input with the control switch in the RESET position. Observe that the amplifier output falls from −100 volts to 0 when the control switch is turned to the POTENTIOMETER SET position.

4.7.5. Servomultipliers. Computer servomechanisms require more maintenance labor than any other computing elements.

Dynamic characteristics as well as static accuracy must be measured in order to assure a high level of computing precision. Frequency-response tests like sine-wave inputs of various frequencies and amplitudes may be performed and the phase and gain recorded, but they are both time-consuming and unnecessary. An adequate test procedure for determining the operational condition of this unit consists of a test of servo performance at a single frequency and amplitude.

Figure 4.7.11 illustrates a typical test circuit for checking servo performance. A triangular wave with a peak-to-peak amplitude of 200 volts and a frequency of 0.05 cps is formed by using an integrating amplifier and relays; a standard oscillator could, of course, be used for the same purpose. The triangular wave drives the servo, while the follow-up-potentiometer output is compared in a series of six summing amplifiers with the servo input and with each of the five multiplying potentiometers. The amplifier outputs are recorded on an oscillograph. Recorder traces illustrating acceptable servo performance are shown in Fig. 4.7.12*a* through *d*. Figure 4.7.12*e* is a recorder trace of a servo which has low gain. Recorder traces of servo potentiometers which are unacceptable with respect to linearity and noise level are shown in Fig. 4.7.12*f* and *g*.

If the servo test indicates low loop gain, the amplifier vacuum tubes and choppers are tested. If these are not at fault, a point-to-point test is made of servo electronics.

If the potentiometer linearity and noise are unacceptable, the servo is dismantled, the servo gears and potentiometer cleaned, and the servo potentiometer inspected with a jeweler's optical glass. After repair, the servo potentiometer is tested by means of a potentiometer linearity tester. Commercially available potentiometer testers contain a motor for oscillating the potentiometer through its range, a standard reference potentiometer, and a comparator for comparing the standard potentiometer with the potentiometer being tested. Such test gear may also be readily constructed in the laboratory.

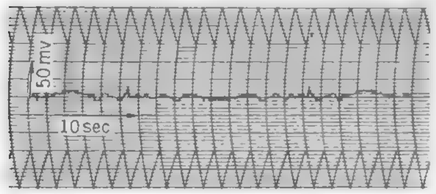


FIG. 4.7.13. Error voltage from servo-resolver test.

4.7.6. Servo Resolvers. Servo resolvers may be tested in a manner similar to that described for the servomultiplier. An indication of servo-resolver dynamic response and loop gain is obtained by comparing the triangular wave input with the voltage on the arm of the feedback potentiometer. To demonstrate the resolver potentiometer linearity and noise, two sine and cosine potentiometers are compared with each other, and time traces are made on the oscillograph. If the recorder trace indicates unacceptable resolver potentiometer performance, a third potentiometer must be used to isolate the faulty unit. Figure 4.7.13 shows a typical recorder trace of an acceptable servo resolver.

4.7.7. Electronic Multipliers. A circuit similar to that described for the servo may be used for testing electronic multipliers. In the case of the servo, of the two variable voltages involved only the servo input is band-pass limited; for the electronic multiplier, the band-pass restrictions on both input variables must be considered. The test circuit of Fig. 4.7.14 was devised for a time-division type of multiplier, but it could be used equally well for testing other electronic multipliers.

A triangular wave with a peak-to-peak amplitude of 200 volts and a frequency of 0.1 cps or higher is used to drive one multiplier input. The other input is first +100 volts and then −100 volts. The multiplier output is filtered appropriately if higher-frequency noise is not significant in the application and interferes with obtaining significant test data. The triangular-wave input is compared with the multiplier output,

and the resulting error voltage is recorded on an oscillograph. Then the X and Y inputs are reversed by means of switch 2, and the test is repeated. Typical test results are shown in Fig. 4.7.15. A 5.0-volt error at the output amplifier corresponds to a 50-mv error in the multiplier.

4.7.8. Coefficient Potentiometers. Since coefficient potentiometers are usually set with the aid of external calibrating devices (Sec. 2.1), it is most frequently sufficient

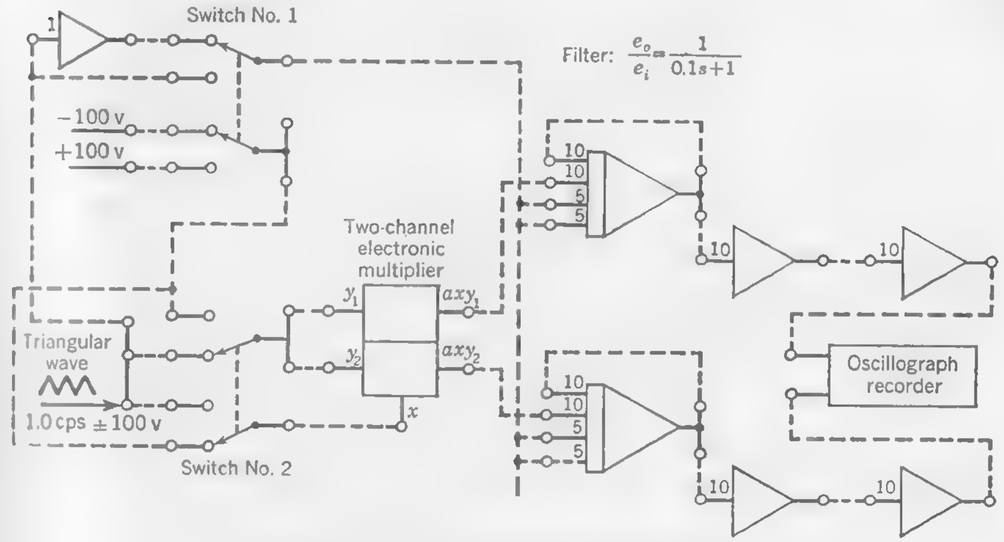


FIG. 4.7.14. Test of electronic multipliers.

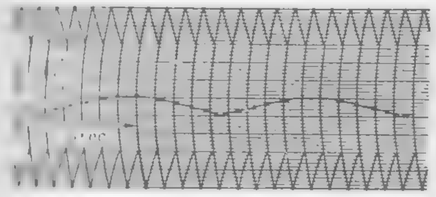


FIG. 4.7.15. Error voltage from electronic multiplier test.

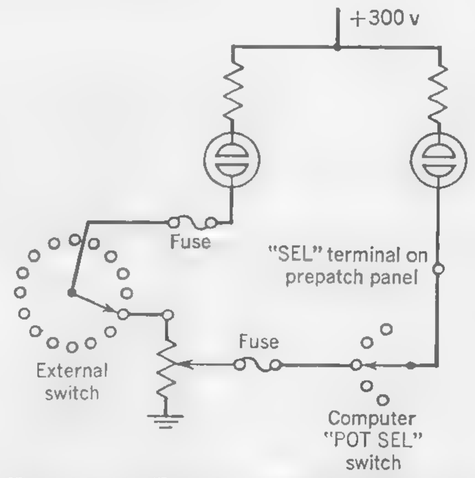


FIG. 4.7.16. Test for potentiometer continuity.

to check potentiometer *continuity* only. Figure 4.7.16 shows a test circuit for this purpose. Potentiometers are selected by a switch on a test board; the NE-51 bulbs will not light if a blown fuse or an open potentiometer exists.

4.7.9. Diodes and Operational Relays. Figure 4.7.17 shows a test circuit which simultaneously tests two diodes and two double-pole double-throw relays. Relay 1 is biased with a negative voltage to achieve a normally closed position for the relay contacts. A positive voltage through contact A overrides the negative bias, which energizes the relay in the opposite direction; then the positive voltage is interrupted and the contacts return to the normally closed position to recycle. The 75-μf capacitor

sets the pulsing rate, and NE2-1 indicates the relay operation. Relay 2 is pulsed at approximately 2 cps through the *B* contacts of relay 1, and NE2-2 indicates the operation of relay 2.

Simultaneously application of the 0.05-cps triangular wave signal to all the diode limiters and pulsing of all positive and negative bias inputs to ground will cause the

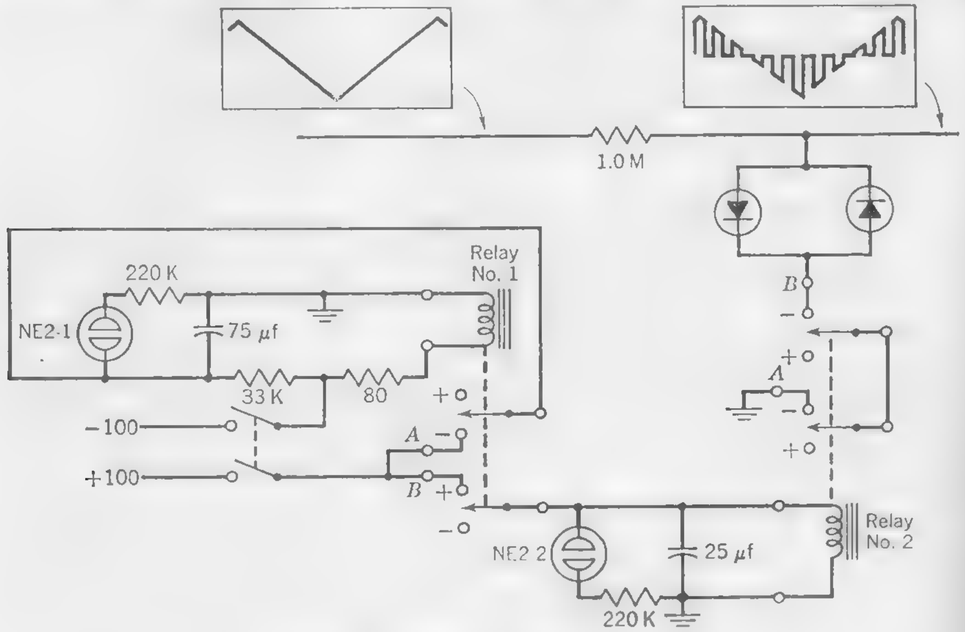
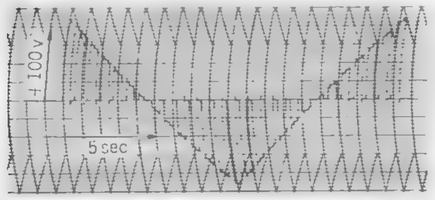
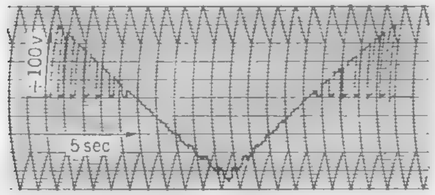


FIG. 4.7.17. Test for diodes and relays.



(*a*) Functional diodes and polarized relay



(*b*) Failed diode

FIG. 4.7.18. Test records of diode-relay circuit.

output waveform shown. Figure 4.7.18 shows output waveforms of a functional and of a defective unit. The advantage of this test circuit is, of course, realized only with a permanently wired test board.

TEST EQUIPMENT

4.7.10. General Considerations. The amount of test equipment required for efficient maintenance will depend on the size and special needs of each computing

laboratory. Ordinarily one can expect initial expenditures for test equipment to range between 5 and 20 per cent of the cost of the computer; the larger percentage applies to smaller laboratories.

4.7.11. Test Benches. A *test bench* for testing and trouble-shooting computer power supplies and computing elements eliminates the need for performing routine maintenance operations at the computer; the computer operating time gained alone will pay quickly for the cost of the test bench. In addition, a well-designed test bench can make the maintenance operations more convenient and efficient.

Desirable features of a test bench include

1. An electronic environment similar to that of the computer
2. Connectors and jumper cables for readily connecting components for test
3. Component inputs, outputs, and other significant test points brought out for convenient monitoring
4. Instrumentation, to include an oscilloscope, digital voltmeter, rms vacuum-tube voltmeter, and a two-channel oscillograph recorder
5. An adequate complement (at least four are preferable) of operational amplifiers with summing and integrating networks; these can be used to generate a triangular wave and to add voltages for servo and electronic multiplier tests
6. Adequate working area on which to rest components and auxiliary instrumentation; a suggested minimum area would be 24 in. deep by 6 ft long
7. Human-factor considerations to govern the test-bench switching and the location of all test-bench instrumentation
8. An oven for precision testing of resistors and capacitors
9. Facilities for trouble-shooting test-bench electronics

4.7.12. Auxiliary Instrumentation. Instruments not generally required on a full-time basis but required periodically, include phase meter, eput meter, test dolly, tube tester, chopper checker, potentiometer cleaner, and potentiometer tester.

(*a*) *Frequency-response Tester.* This unit is used to measure the gain and phase of computing components. The following frequency ranges are typical:

Servos	0.05–20 cps
Electronic multipliers	1.0–100 cps
Operational amplifiers	0.1–100 kc

(*b*) *Phase Meter.* This unit is less expensive than a general-purpose frequency-response tester and is particularly useful for quick checks of amplifier performance. A resolution of 0.1 deg of phase shift is desirable for this application.

(*c*) *Eput* (*events-per-unit-time*) *Meter.* This unit provides a capability for a precision time measurement between electronic pulses or voltage steps. It is useful for measuring the relay make-to-break time and the operational-amplifier integration rate.

(*d*) *Test Dolly.* In the larger laboratories commonly used test instrumentation is mounted on a wheeled dolly. A test dolly might comprise a vacuum-tube voltmeter, an oscilloscope, a digital voltmeter (with an internal standard cell), and an eput meter.

(*e*) *Tube Tester.* The card-insert type of tube tester, which permits the user to establish his own standards for acceptability of tubes, is preferred.

(*f*) *Chopper Checker.* It is desirable to construct or buy a unit that will contain the necessary power supplies to permit convenient testing of all mechanical choppers. It is undesirable to have this unit independent of the test bench so that the test bench is not tied up for this purpose.

(*g*) *Potentiometer Cleaner and Potentiometer Linearity Tester.* A major problem area in the analog-computer maintenance laboratory is to maintain servo-potentiometer noise and linearity at an acceptably low level. The best servo potentiometers have a normal life at high performance of about 3 months, depending on environment and use. Oil-immersed potentiometers appear to have a somewhat higher useful life. The cost of repairing servo potentiometers is high, and delays in factory repairs necessitate a larger spares inventory. Consequently, it is often more economical to perform the potentiometer repair within the computer laboratory.

Two instruments which are essential, if serious consideration is given to the repair

of servo potentiometers within the maintenance laboratory, are a cleaner, to clean potentiometers and servo gears, and a potentiometer linearity tester.

Various suitable types of *potentiometer cleaners* are commercially available. One type effects cleaning by means of a high-pressure spray. Another comprises a glass container with an agitator.

The typical *potentiometer linearity tester* mounts the potentiometer under test on a motor shaft. The potentiometer is oscillated through its angular range while its output is compared with the output of a standard potentiometer. This tester permits check-out of a potentiometer before it is assembled on the servo.

4.7.13. Test Boards. A *test board* prewired with the circuits described in Arts. 4.7.2 to 4.7.9 and plugged into the computer patchbay will reduce the number of manual operations that the technician must perform to obtain the required test data for a complete analog computer.

To provide the test circuits necessary to check a complete computing console within a single test board requires considerable ingenuity and rather complex switching circuitry. Consequently many computing laboratories have taken the more straightforward approach of using several test boards for various test functions.

MAINTENANCE PROCEDURES

4.7.14. Types of Maintenance Operations. The maintenance tasks required within the electronic-analog-computing facility can be divided into the following three classes. *Preventive maintenance* includes planned periodic marginal and functional testing of computing system and components. *Routine maintenance* consists of the regular repair of failed components at the test bench. *Emergency maintenance* includes trouble shooting of computing system and replacement of components during a computer problem period.

4.7.15. Preventive Maintenance. The type and frequency of preventive maintenance required are, to an extent, a function of how the computing laboratory is utilized; whether the laboratory is operated on an open- or closed-shop basis; whether single- or dual-shift operation is in effect; and whether the computer is used as a special-purpose device with problem wiring unchanged for several months at a time, or as a general-purpose computer with frequent problem changes. Some large computing laboratories perform a daily test-board check of most or all computing elements. These tests are performed by a team of technicians each handling one test board and proceeding in turn to test the equipment within each computing console. The tests are such that potentiometer or function generator settings need not be adjusted; the test boards are used in a definite order, since some of the test circuits used assume proper functioning of other computing components not under test. In the course of the tests, the technician replaces the failed components from a spares supply. The failed component is stored to be repaired after the system tests have been completed. The average time spent on this daily check is about 1½ hr. The laboratories which perform these tests are generally operated on an open-shop basis, with problem changes occurring at a relatively high frequency. A few of these laboratories are operated on a double-shift basis with different problems scheduled for each shift. When a laboratory is operated open-shop, personnel using the equipment do so only occasionally, and the mortality rate on fuses and wear on equipment will be greater. Hence, for this type of operation, it may be expected that a greater need would exist for a daily check of equipment.

Some laboratories use test boards to perform daily functional checks of potentiometers, input resistors, and amplifiers only. These tests usually require about 30 min. Other components, such as servo electronic multipliers, diodes, and relays, are checked weekly or semimonthly. In still other computing laboratories, complete tests of the computing system are performed only between problems. Components failing between tests are handled on an emergency basis. Between periodic checks one may further check the machine through frequent reference to a standard test-problem solution. In some instances, a solution of the problem on a digital computer is used as a reference standard. It should be noted that only the equipment used in the test problem would be checked by this method, and this to an extent which is dependent on its significance in the problem solution.

Other facilities, which are operated essentially as special-purpose computers in which problems extend for several months or longer, rely mainly on the standard-problem-solution method of checking and perform complete functional tests only at wide time intervals.

Preventive maintenance is a desired objective of most computing laboratories. However, the only preventive maintenance in extensive use involves tests of marginal performance. Some of these tests have already been described. A few laboratories, however, remove components from the system on a regular schedule in order to perform a more comprehensive evaluation than is possible at the computer. This evaluation includes the testing of all tubes and choppers in the components which are pulled. It should be noted that this approach to preventive maintenance is subject to some controversy; conceivably, more harm than good is done by unnecessary removal of components.

4.7.16. Routine Maintenance. Routine maintenance consists of the daily trouble shooting of plug-in computer components. Normally this consists of testing tubes and choppers and point-to-point testing of units for voltage and impedance. The test-bench facility is designed to expedite this routine maintenance.

4.7.17. Emergency Maintenance. Computer plug-in components which fail during use are normally replaced with functional units. If the failure is not within a plug-in component, computer down time may be required to effect the repair. If it is possible to continue the computation by using other components, the repair is delayed until the scheduled maintenance period. Cards located on the computing console are used for itemizing such repairs to assure prompt attention when scheduled maintenance is performed.

Occasionally, operating personnel may suspect a component failure but are unable to isolate the failure. In such instances, the study board may be removed and test boards used to check the operating condition of the computer.

Failures which occur during operating periods are costly in terms of loss of time and because they distract maintenance personnel from routine maintenance duties. Consequently, there is a high premium on marginal testing and preventive-maintenance methods which can increase the mean error-free operating time.

MAINTENANCE RECORDS

4.7.18. Function and Significance of Records. The statistical facts obtained from good maintenance records can be invaluable for locating problem areas and for taking corrective action. On the other hand, a poorly organized record system can cause needless expenditure of valuable maintenance time. The procedure for record keeping must be well organized so that a minimum percentage of the technician's time is spent in evaluating the results of his efforts.

Maintenance records should provide at least the following information:

1. Proper identification of the individual unit repaired
2. Symptoms observed
3. Corrective action taken
4. Name of the individual who performed the repair
5. Date when the repair was effected

4.7.19. Examples. On a well-thought-out failure report sheet (Convair Aircraft Corporation, San Diego Division),[2] typical types of failure symptoms for the various computing components are tabulated on the front with space left for comments. The back of the failure report is used to indicate the nature of the repair. Details required for statistical analyses are called out and minimize thought and comment required by the technician. Because of the nature of the maintenance procedure the failure report (front) and repairman's report (back) are not in general made out by the same technicians. The form is such that data may be readily reduced for summary and evaluation by a digital computer.

In addition to the failure report, it is desirable to maintain a Kardex type of record file on all computer components. This type of file provides the maintenance history of a given unit and is analogous to the record file that a doctor keeps on each patient. This file is accumulated through reduction of data tabulated on failure and repairman's reports; space is provided for a minimum of notes on symptoms and repair, date, location of equipment in system, and the individual who performs the repair. If the same symptoms and repairs occur at too frequent periods, a need for more permanent remedial action would be indicated.

At the Space Technology Laboratories (Los Angeles, Calif.), a file system is employed[3] for their preventive-maintenance programs to assure that all components are checked at precstablished periods. A dated file on all components is kept so that units are inspected as their cards come to the front of the dated file. Each file card identifies the component and indicates the number of weeks between periodic inspections, the date of the last inspection, and the technician's initials.

It is desirable to file the oscillograph-recorder test data on all servomultipliers, servo resolvers, and electronic multipliers, at least for the last three tests performed on the units. A notebook with transparent pockets, similar to a stamp collector's album, is ideal for this purpose. This book may be utilized by computer users to select optimum potentiometers or components, and by maintenance personnel for reviewing the past history of each unit.

4.7.20. Equipment Identification. It is essential that the computing system and components be clearly and permanently labeled for proper identification, in particular:

1. Component slots must be referenced to the computer patchbay.
2. A positive identification number (in addition to the manufacturer's serial number) must be assigned to each component and computer rack.

The first identification is necessary for convenience in maintaining and using the computing system. The second identification is required for records of inventory and component maintenance histories.

SPARE PARTS

4.7.21. Spare-parts Supply. To assure efficient repairs, a well-organized and complete spares supply must be available within the maintenance laboratory. Adequate storage facilities, such as cabinets, drawers, and bins, should be provided so that the necessary large variety of spare tubes, choppers, resistors, capacitors, potentiometers, fuses, switches, etc., can be stored and properly identified.

To assure a continuous supply of spares one may have a recorder list posted at a convenient location and instruct the technicians to enter items in low supply. In contrast to this, one large computing laboratory keeps a running account of spares supplies of critical items with the aid of an IBM computer.

Spares should be stored for all plug-in-type computing components, amplifiers, servos, and power supplies. Experience indicates that an adequate spares supply of operational amplifiers would be two units, or 5 per cent of the total number of amplifiers in the computing system, whichever is greater. For servos, it is desirable to have at least one spare unit of each servo type, or 15 per cent of the servos in the computing system, whichever is greater.

References

1. Hamer, H., and J. D. Kennedy: Testing Operational Amplifiers, *J. Assoc. Computing Machinery*, **2**: 92 (April, 1955).
2. Horwitz, R. D.: Maintaining an Analog Computing System, *Electronic Inds. Tele-Tech.*, December, 1958.
3. Sykes, R. P.: A Preventive Maintenance Program for Large General-purpose Electronic Analog Computers, Space Technology Laboratories, Los Angeles, Calif., 1958.

Section 5

ELECTRONIC ANALOG COMPUTERS: SIGNIFICANT APPLICATIONS

By

ARTHUR I. RUBIN, *Director, Princeton Computation Center, Electronic Associates, Inc., Princeton, N.J.*

VICTOR B. COREY, *President, Palomar Scientific Corporation, Palo Alto, Calif., and Vice-President, United Control Corporation, Seattle, Wash. (formerly Technical Director, Donner Scientific Company, Concord, Calif.).*

JOHN McLEOD, *Design Specialist, General Dynamics/Astronautics, San Diego, Calif.*

GRANINO A. KORN, *Professor of Electrical Engineering, University of Arizona, Tucson, Ariz., and* **THERESA M. KORN.**

LOUIS BAUER, *Director, Project CYCLONE, Reeves Instrument Corporation, Garden City, N.Y.*

CHARLES W. WORLEY, *Consulting Engineer (formerly Market Development Manager, Electronic Associates, Inc., Long Branch, N.J.).*

E. MORRISON, *Assistant Director, Applied Research Laboratory, University of Arizona, Tucson, Ariz.*

VINCENT C. RIDEOUT, *Professor of Electrical Engineering, University of Wisconsin, Madison, Wis.*

R. M. HOWE, *Professor of Aeronautical Engineering, University of Michigan, Ann Arbor, Mich.*

L. D. KOVACH, *Design Specialist, Engineering Computing Group, Douglas Aircraft Company, El Segundo Division, and Professor of Mathematics, Pepperdine College.*

H. F. MEISSINGER, *Assistant Manager, Systems Development Group, Hughes Aircraft Company, Culver City, Calif.*

R. P. WASHBURN, *Analyst, Marketing Research, Computer Department, General Electric Co., Phoenix, Ariz.*

CONTENTS

PART 1: ANALOG-COMPUTER PROGRAMMING, SCALING, AND PROBLEM PREPARATION

Arthur I. Rubin

Block Diagrams and Scaling for Basic Operations

5.1.1. Block-diagram Symbols..... 5–5
5.1.2. Basic Circuits: Addition and Multiplication by Constant Coefficients..... 5–5
5.1.3. Basic Circuits: Integration and Initial Conditions..... 5–5
5.1.4. Basic Circuits: Multiplication with Servomultipliers..... 5–5
5.1.5. Basic Circuits: Use of Electronic Multipliers..... 5–8
5.1.6. Division and Root Extraction by Implicit Computation.... 5–8
(*a*) Two-quadrant Division with Servomultipliers..... 5–8
(*b*) Indeterminate Quotients and Four-quadrant Division..... 5–9
(*c*) Square-root Extraction..... 5–12
(*d*) Use of Electronic Multipliers.. 5–12

Scaling a Problem

5.1.7. General..... 5–12
5.1.8. Amplitude Scaling..... 5–12
5.1.9. Example: Solution of a Linear Differential Equation..... 5–13
5.1.10. Example: Solution of a Nonlinear Differential Equation.. 5–15
5.1.11. Other Scaling Methods..... 5–16
5.1.12. Time Scaling..... 5–17

Use of Function Generators

5.1.13. General Rules..... 5–18
5.1.14. Nonlinear Analytic Approximations..... 5–18

Setup Sheets and Checking Procedures

5.1.15. Setup Sheets..... 5–19
5.1.16. Checking Procedures..... 5–19
5.1.17. Records and Summary of Problem-solving Procedure... 5–20

PART 2: MISCELLANEOUS APPLICATIONS (TABLE)..... 5–23

Victor B. Corey

PART 3: ELECTRONIC-ANALOG-COMPUTER TECHNIQUES FOR THE DESIGN OF SERVO SYSTEMS

John McLeod

Analog Representation of Servo Systems

5.3.1. Servo System and Computer Analogy..... 5–35
5.3.2. Linear Servomechanisms..... 5–35
5.3.3. Nonlinear Systems..... 5–35
5.3.4. Sampled-data Systems..... 5–37

Performance Criteria

5.3.5. Evaluation of Computer Data. 5–37
5.3.6. Steady-state Displacement Error, Velocity Error, and Acceleration Error..... 5–37
5.3.7. Other Measures of Servo Performance..... 5–40

PART 4: SPECIAL CIRCUITS FOR DYNAMICAL-SYSTEM STUDIES

Granino A. Korn and Theresa M. Korn

5.4.1. Representation of Nonlinear Transfer Characteristics..... 5–42
5.4.2. Limit Stops and Backlash..... 5–43

PART 5: AIRCRAFT, AUTOPILOT, AND MISSILE PROBLEMS

Louis Bauer

Rigid-body Motion of Aircraft

5.5.1. Flight Equations and Simplifying Assumptions..... 5–49
5.5.2. Coordinate Systems..... 5–50
5.5.3. Typical Three-dimensional Flight Equations..... 5–51
5.5.4. Discussion of Equations and Alternate Formulation..... 5–52
5.5.5. Computer Flow Diagram.... 5–54
5.5.6. Typical Simplified Flight Equations..... 5–55
5.5.7. Limitations. Use of Direction Cosines..... 5–59

Control Systems

5.5.8. Airplanes..... 5–60
5.5.9. Missiles..... 5–60
5.5.10. General Procedures..... 5–60
5.5.11. Some Special Techniques.... 5–60

Non-rigid-body Motion

5.5.12. Treatment of Aeroelasticity.. 5–62

Solution of Large Problems

5.5.13. Special Precautions..... 5–63

PART 6: PARTIAL SYSTEM TESTS AND FLIGHT TABLES

Louis Bauer

Introduction

5.6.1. Partial System Tests..... 5–64

Partial System Tests Including a Human Operator

5.6.2. Flight Simulators..... 5–64
5.6.3. Missile-guidance Trainers.... 5–65
5.6.4. Other Training Devices..... 5–65

Component and Subsystem Tests

5.6.5. Flight Tables..... 5–65
5.6.6. Roll Tables (Single-axis Flight Tables)..... 5–66
5.6.7. Three-axis Flight Tables..... 5–66
5.6.8. "Gimbal Computation" in Flight Tables. Gyro-controlled Flight Tables..... 5–67
5.6.9. Tachometer-controlled Flight Tables..... 5–68
5.6.10. Position-controlled Flight Tables..... 5–69
5.6.11. Installation..... 5–70
5.6.12. Other Partial System Tests.. 5–70
5.6.13. Discussion..... 5–70

PART 7: PROCESS-CONTROL APPLICATIONS

Charles W. Worley

Introduction

5.7.1. Process-control Systems..... 5–71
5.7.2. Process Models and Test Data 5–71

Analog-computer Representation of the Control System

5.7.3. Linearized Representation of Control-system Components. 5–72
5.7.4. Representation of a Typical Process-control System..... 5–75

Process Models and Their Representation

5.7.5. Representation of Process Dynamics..... 5–78
5.7.6. Some Process Models and Computer Representations... 5–78
5.7.7. A Process Model Involving Partial Differential Equations: Representation of a Heat Exchanger..... 5–80
5.7.8. Simulation of a Distillation Column..... 5–80

Processes Involving Chemical Reactions

5.7.9. Introduction..... 5–82
5.7.10. Analog-computer Representation of a Chemical Process... 5–83
5.7.11. More General Reaction-rate Equations..... 5–85

PART 8: NUCLEAR-REACTOR SIMULATION

E. Morrison

5.8.1. Introduction..... 5–87
5.8.2. Representation of Nuclear Kinetics..... 5–87
5.8.3. Representation of Poison Generation..... 5–90
5.8.4. Representation of Heat Transfer..... 5–91

PART 9: RANDOM-PROCESS STUDIES

Vincent C. Rideout

Basic Definitions and Formulas

5.9.1. Introduction..... 5–94
5.9.2. Description of Random Processes..... 5–94
5.9.3. Ensemble Averages and Correlation Functions..... 5–94
5.9.4. Effect of Linear Operations.. 5–94
5.9.5. Ergodic Processes and Time Averages..... 5–95
5.9.6. Time-correlation Functions.. 5–95
5.9.7. Correlation Functions and Spectra..... 5–95
5.9.8. Effect of Linear Operations on Spectra..... 5–95

Analog-computer Techniques

5.9.9. Analog Computation of Statistical Quantities: Survey... 5–96
5.9.10. Noise Generators and Spectrum Shaping..... 5–96
5.9.11. Scaling Problems and Prewhitening..... 5–96

Determination of Correlation Functions and Delay Error

5.9.12. Analog Measurement of Correlation Functions and Delay Error: Stationary Case..... 5–96
5.9.13. Errors in Time-correlation Measurements..... 5–98
5.9.14. "Mathematical" Computation of Correlation Functions and Delay Error from Known Correlation Functions: Stationary Case..... 5–98
5.9.15. System Analysis for Nonstationary Problems..... 5–99
5.9.16. The Adjoint-system Method. 5–101
5.9.17. A Special Case..... 5–102

Measurement of Spectral Densities

5.9.18. Measurement of Power Spectra..... 5–102
5.9.19. Measurement of Cross-spectral Density..... 5–103
5.9.20. Errors in Power-spectrum Measurements..... 5–104
5.9.21. Fourier Inversion..... 5–105

Analog Computation of Probability Estimates

5.9.22. Probability-distribution Analyzers..... 5–107
5.9.23. Probability-density Analyzers 5–107
5.9.24. Generation of Random Voltages of Given Spectra and Probability Densities..... 5–108

PART 10: SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

R. M. Howe

Introduction

5.10.1. Requirement for the Solution of Partial Differential Equations..... 5–110
5.10.2. Example Types of Partial Differential Equations..... 5–110
5.10.3. Importance of Error Analysis..... 5–110

Solution by Separation of Variables

5.10.4. Basic Equation for Heat Flow..... 5–110
5.10.5. Separation of Variables... 5–112
5.10.6. Computer Circuit for the Boundary-value Problem... 5–113
5.10.7. Generation of the Fourier Coefficients..... 5–114
5.10.8. Analog Computation of the Complete Series Solution... 5–115
5.10.9. Additional Eigenvalue Problems..... 5–116
5.10.10. Use of Normal Coordinates to Represent a Continuous System..... 5–118
5.10.11. Appraisal of the Method of Separation of Variables..... 5–119

Solution by Difference Techniques

5.10.12. One-dimensional Heat Flow 5–119
5.10.13. Estimated Accuracy of the Difference Method Using a Taylor-series Expansion.... 5–122
5.10.14. Estimated Accuracy of the Difference Method Using Separation of Variables..... 5–123
5.10.15. Higher-order Central Differences and Backward Differences..... 5–123
5.10.16. Heat Flow with Mixed Boundary Conditions..... 5–125
5.10.17. Two-dimensional Heat Flow 5–125
5.10.18. Heat Flow in Cylinders..... 5–126
5.10.19. Spherical and Other Boundary Shapes..... 5–128
5.10.20. Change of Variables to Group Stations Unevenly..... 5–128
5.10.21. Solution of Heat-flow Problems in Semi-infinite Media. 5–128
5.10.22. Nonlinear Heat-flow Problems..... 5–128
5.10.23. Problems with Moving Boundary Conditions..... 5–129
5.10.24. Solution of the Wave Equation..... 5–129
5.10.25. Transverse Displacement of Thin Beams..... 5–129
5.10.26. Displacement of Thick Beams Including Transverse Shear Effects..... 5–130
5.10.27. Beams with Damping..... 5–130
5.10.28. Stability of Beam Circuits.. 5–130
5.10.29. Representation of Plates and Built-up Structures..... 5–131
5.10.30. Comparison of Passive-circuit Methods and Electronic-analog-computer Techniques 5–131

PART 11: SOLUTION OF ALGEBRAIC EQUATIONS, LINEAR PROGRAMMING, AND PARAMETER OPTIMIZATION

L. D. Kovach and H. F. Meissinger

Algebraic Equations

5.11.1. Linear Algebraic Equations and Eigenvalue Problems.. 5–133
5.11.2. Evaluation of Determinants. 5–138
5.11.3. Inversion of Matrices..... 5–138

Analog-computer Methods for Finding Roots of Polynomial Equations

5.11.4. General..... 5–139
5.11.5. Scan Methods..... 5–141

Null and Track Methods

5.11.6. Relaxation Method..... 5–144
5.11.7. Method of Steepest Descent. 5–145
5.11.8. Root-locus Tracking..... 5–146
5.11.9. Miscellaneous Methods..... 5–148

Linear Programming

5.11.10. Theory..... 5–149
5.11.11. Analog-computer Solution.. 5–151

Special Techniques for Parameter Optimization

5.11.12. Parameter Optimization.... 5–151
5.11.13. The Method of Parameter Influences..... 5–153

PART 12: ECONOMIC-DISPATCH COMPUTERS FOR POWER SYSTEMS

R. P. Washburn

5.12.1. Introduction..... 5–155
5.12.2. Problem Statement..... 5–155
5.12.3. Optimum Scheduling with Negligible Transmission Losses 5–155
5.12.4. Complete Economic-dispatch Computers..... 5–159
5.12.5. EDC's as Control Systems... 5–159

PART 1: ANALOG-COMPUTER PROGRAMMING, SCALING, AND PROBLEM PREPARATION

By ARTHUR I. RUBIN

BLOCK DIAGRAMS AND SCALING FOR BASIC OPERATIONS

5.1.1. Block-diagram Symbols. Various systems of block-diagram symbols are [illegible] and illustrated in Sec. 1.2. This section will employ the Electronic Associates notation, which is used in many analog-computer installations in the United States, as well as the more recent IRE-preferred symbols.

5.1.2. Basic Circuits: Addition and Multiplication by Constant Coefficients. Figure 5.1.1*a* shows the relation of summing-amplifier input gain to computer scaling for typical summation circuits. Input gains of 1, 5, and 10 are assumed, but the circuits shown are perfectly general, since the desired input gains can be obtained with external resistors. Figure 5.1.1*b* illustrates the use of feedback for obtaining fractional gain, and Fig. 5.1.1*c* and *d* shows the use of coefficient-setting potentiometers.

5.1.3. Basic Circuits: Integration and Initial Conditions. Figure 5.1.2*a* shows a number of circuits implementing integration with respect to time together with addition and multiplication by constant coefficients.

Frequently overlooked is the necessity for inserting the correct sign of an initial condition (in relation to the indicated sign of the integrator output), as well as the scaling of the initial condition to correspond with the indicated scale shown for the output of the integrator. Various possibilities of indicating the scale factor and sign of the reference on the initial-condition potentiometer are shown in Fig. 5.1.2*b*.

5.1.4. Basic Circuits: Multiplication with Servomultipliers. Setup of computer servomechanisms involves connections to the servo amplifier and follow-up potentiometer cup (*follow-up excitation*) and to the multiplying potentiometer cups. Figures 5.1.3 and 5.1.4 illustrate the three follow-up excitations commonly used for four-quadrant and two-quadrant multiplication. Note that four-quadrant multiplication requires both signs of the multiplicands (y and $z/10$) for each multiplier cup, whereas in two-quadrant multiplication only one sign of the multiplicand is required for each multiplier cup. This fact can lead to the saving of equipment by a proper choice of variables for the roles of multiplier and multiplicand.

Load-equalizing resistors (Sec. 3.2), which ensure identical attenuation in the follow-up and multiplying potentiometers, are indicated in Fig. 5.1.3. Three possible block-diagram notations are shown.

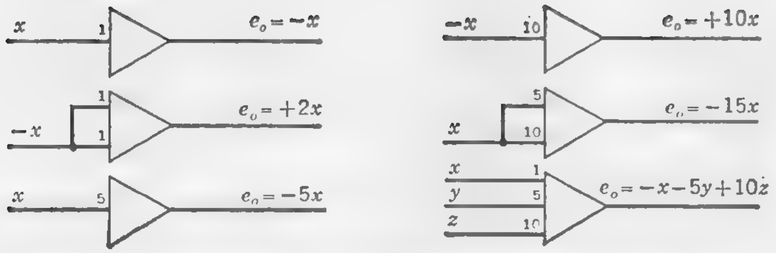


FIG. 5.1.1*a*. Addition and multiplication by integral coefficients.

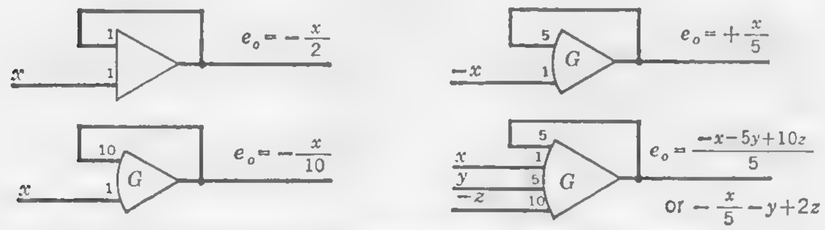


FIG. 5.1.1*b*. Addition and multiplication by rational fractions.

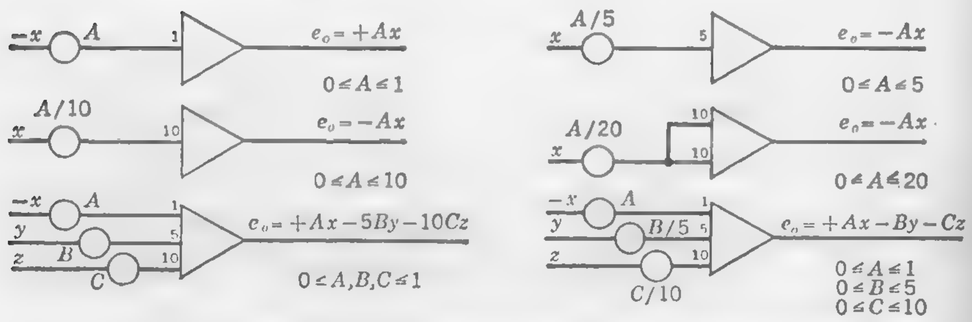


FIG. 5.1.1*c*. Addition and multiplication by arbitrary coefficients.

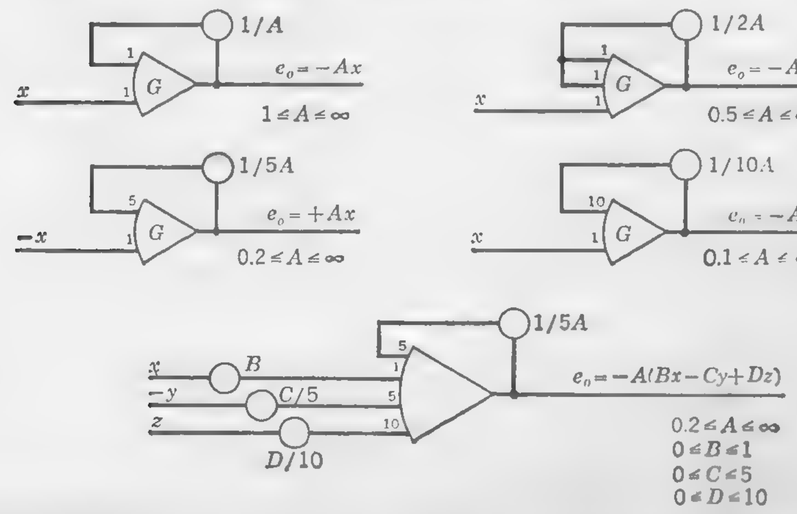


FIG. 5.1.1*d*. Addition and multiplication by arbitrary coefficients—potentiometer in feedback path. Note that essentially infinite gain can be obtained in this manner, but zero gain cannot be obtained.

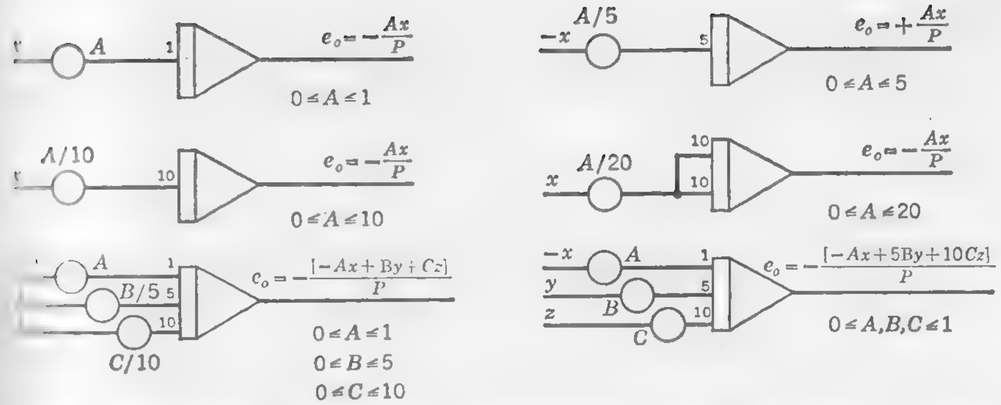


FIG. 5.1.2*a*. Integration, addition, and multiplication by constant coefficients. Note that $e_o = \int_0^t x\,dt$ if $e_o(0) = 0$.

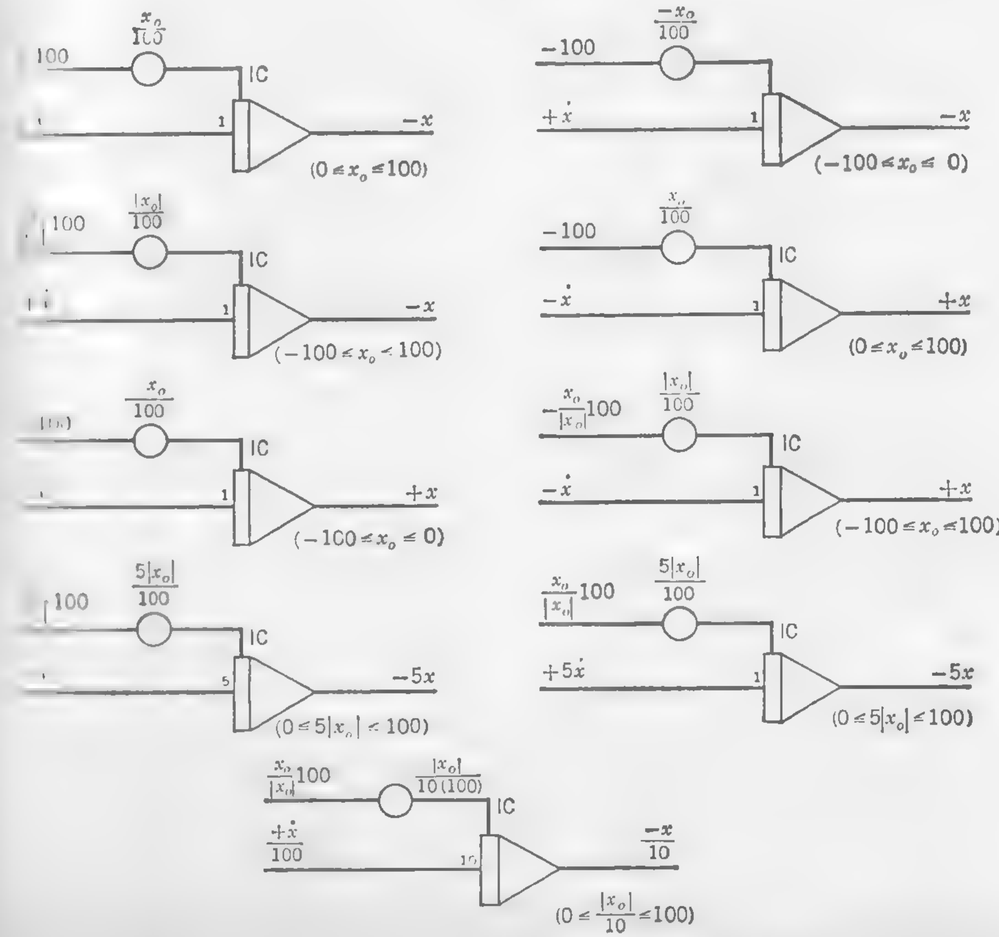


FIG. 5.1.2*b*. Scaling of initial conditions for integrators which invert the sign of their initial-condition input. The integrators in most multipurpose electronic analog computers are of this type (see also Sec. 4.1).

5.1.5. Basic Circuits: Use of Electronic Multipliers. Figure 5.1.5 shows setup and scaling for multiplication and for division with electronic multipliers having a built-in division mode (Sec. 3.3). The use of time-division multipliers with slave channels (Sec. 3.3) is also indicated.

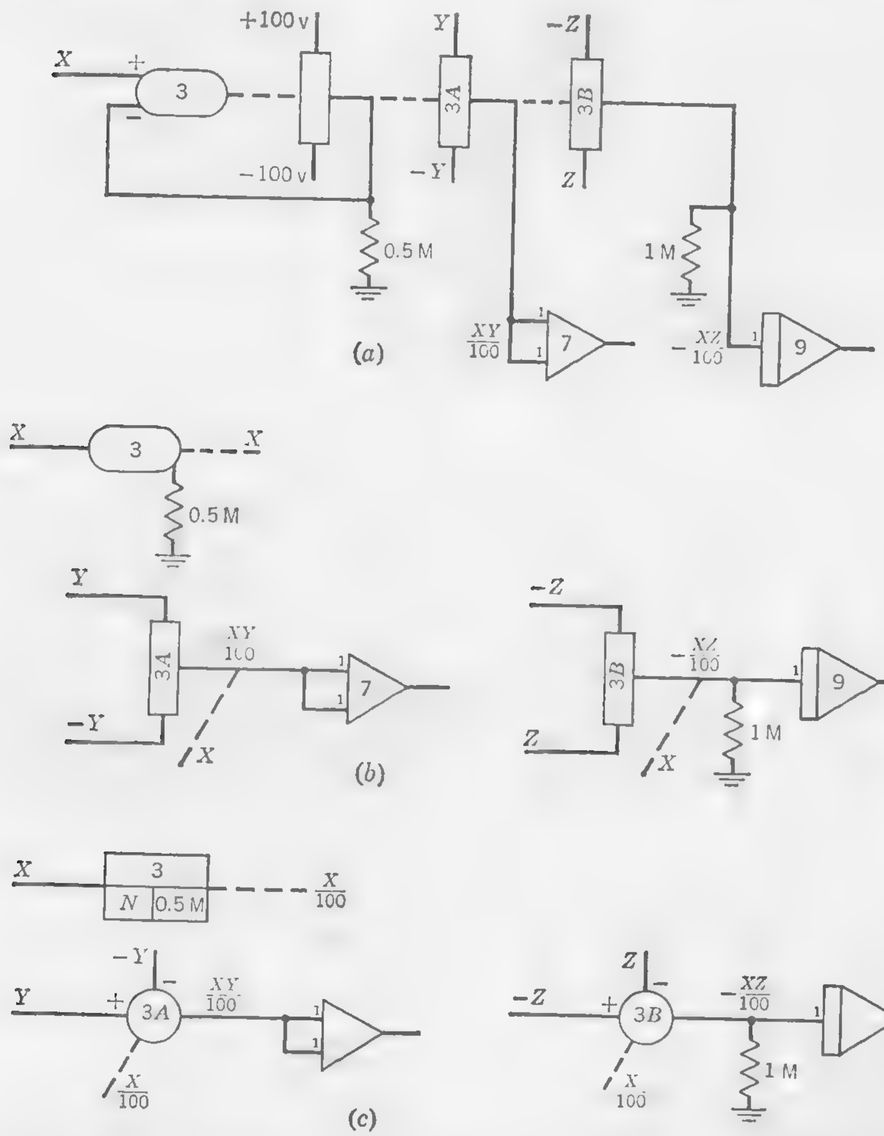


FIG. 5.1.3. Four-quadrant servo multiplication in IRE standard notation (*a*), shortcut notation (*b*), and Electronic Associates notation (*c*). Note the load-equalizing resistors; in the case shown, the servo amplifier does not load the follow-up potentiometer. The follow-up excitation is "normal" and has been omitted in (*b*). See text. In Figs. 5.1.3*a* and *b*, (*a*), the *upper* end of each potentiometer cup corresponds to the end marked + in the Electronic Associates notation (Fig. 5.1.3*c*). Addition of the + and − symbols to the IRE and shortcut notations might be helpful.

5.1.6. Division and Root Extraction by Implicit Computation. (*a*) *Two-quadrant Division with Servomultipliers.* If a multiplier cup is placed in the feedback path of a high-gain amplifier, division of the input variable to the high-gain amplifier by the multiplier is accomplished. Circuits and scaling are illustrated in Fig. 5.1.6, using Electronic Associates notation. Note that only two-quadrant division can be performed, since the divisor is not allowed to be zero. This latter condition corresponds to an infinite gain for the high-gain amplifier.

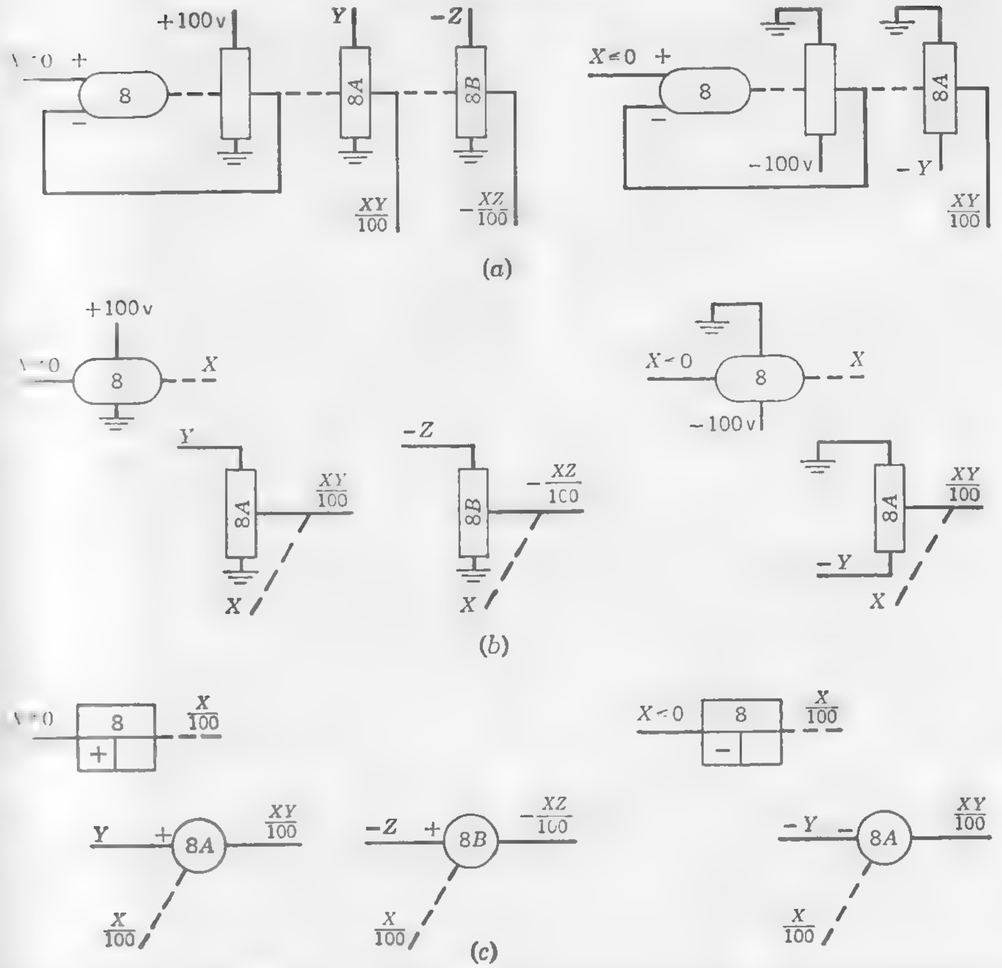


FIG. 5.1.4. Two-quadrant servo multiplication in IRE standard notation (*a*), shortcut notation (*b*), and Electronic Associates notation (*c*). Load-equalizing resistors may be added as necessary.

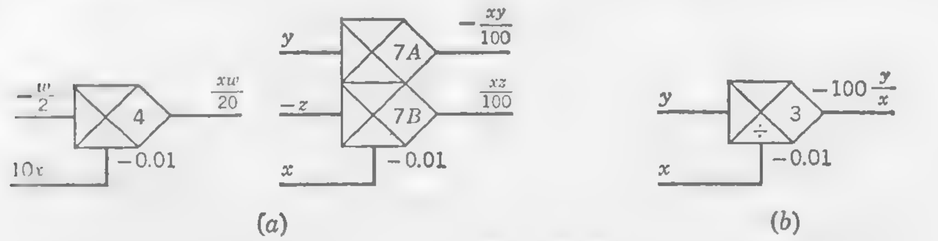


FIG. 5.1.5. Four-quadrant multiplication (*a*) and two-quadrant division (*b*) with electronic time-division multipliers having a built-in division mode (see also Sec. 1.2).

(*b*) *Indeterminate Quotients and Four-quadrant Division.* Some differential-equation problems (e.g., trajectory problems, Bessel's differential equation) involve quotients of the form $X(t)/Y(t)$ where both $X(t)$ and $Y(t)$ start from zero at $t = 0$. Figure 5.1.7 shows a circuit which represents the indeterminate form by $(X + Ae^{-\alpha t})/(Y + Be^{-\alpha t})$ where $A/B = X(0)/Y(0)$, and $1/\alpha$ is a small time constant.

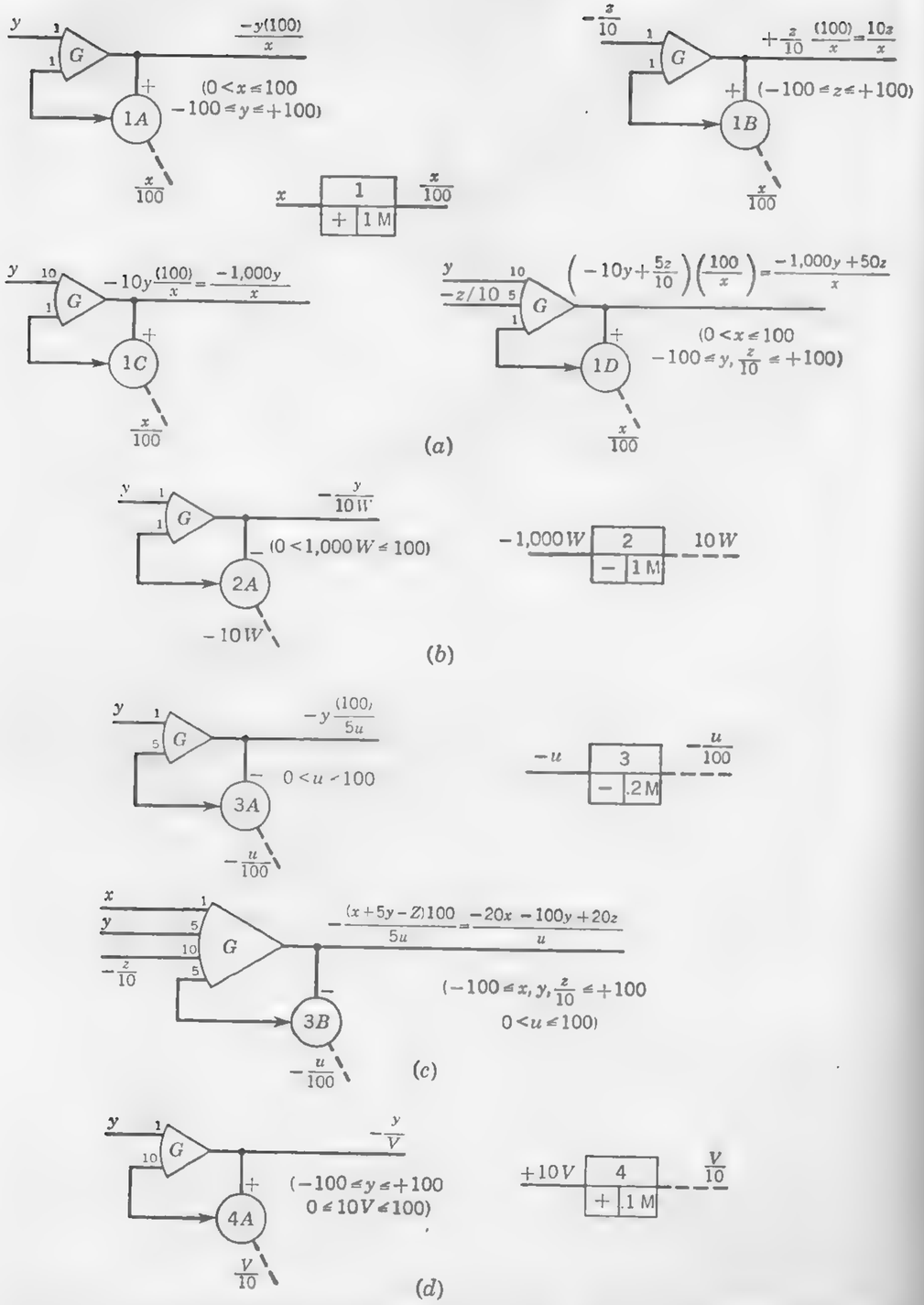


FIG. 5.1.6. Two-quadrant division with servomultipliers. Electronic Associates notation; note provisions for load-equalizing resistors.

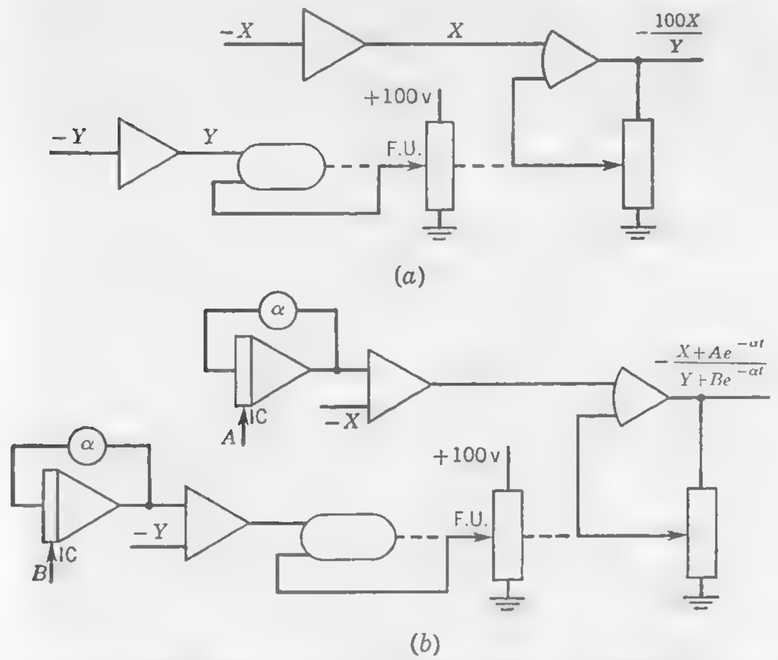


FIG. 5.1.7. Two-quadrant-division circuit in IRE-preferred notation (*a*), and modified circuit (*b*) permitting computation of quotients X/Y in cases where both X and Y start from zero at $t = 0$. A and B are chosen so that $A/B = \lim_{t\to 0} (X/Y)$, and $1/\alpha$ is a small time constant. (*Contributed by L. Bauer, Reeves Instrument Corp.*)

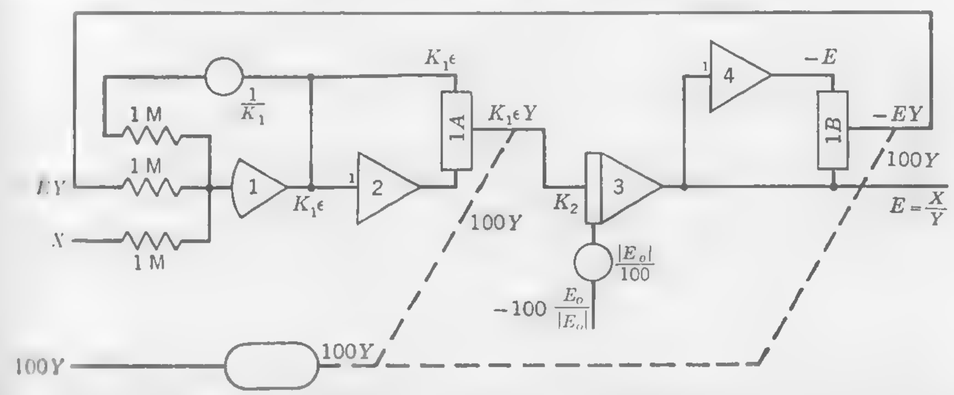


FIG. 5.1.8. Four-quadrant division by the method of steepest ascents. K_1 and K_2 are usually of the order of 100 or greater. An initial condition set on the integrator produces the correct initial value E_0 of E. If Y lies only in one quadrant, then either amplifier 2 or amplifier 4 can be eliminated. Amplifier 2 can be omitted if Y is positive only, and amplifier 4 can be omitted if Y is negative only; the servo follow-up excitation is then either "plus" or "minus," respectively.

The more general four-quadrant division circuit of Fig. 5.1.8 permits $X(t)$ and $Y(t)$ to go through zero together at any time t during the computation, provided that $\lim X(t)/Y(t)$ is finite. $E = X/Y$ is obtained as solution of the differential equation

$$\frac{dE}{dt} = -K_1K_2\epsilon\frac{\partial\epsilon}{\partial E} = -K_1K_2\epsilon Y$$

where $\epsilon = EY - X$, and K_1K_2 is a large positive constant.

(*c*) *Square-root Extraction.* The square-root operation is obtained on the analog computer by a squaring operation in the feedback path of a high-gain amplifier. This is another example of the use of inverse operations in the feedback path to obtain a mathematical operation. Figure 5.1.9 shows typical scaled circuits in Electronic Associates notation. Analogous techniques permit computation of cube roots and other fractional powers.

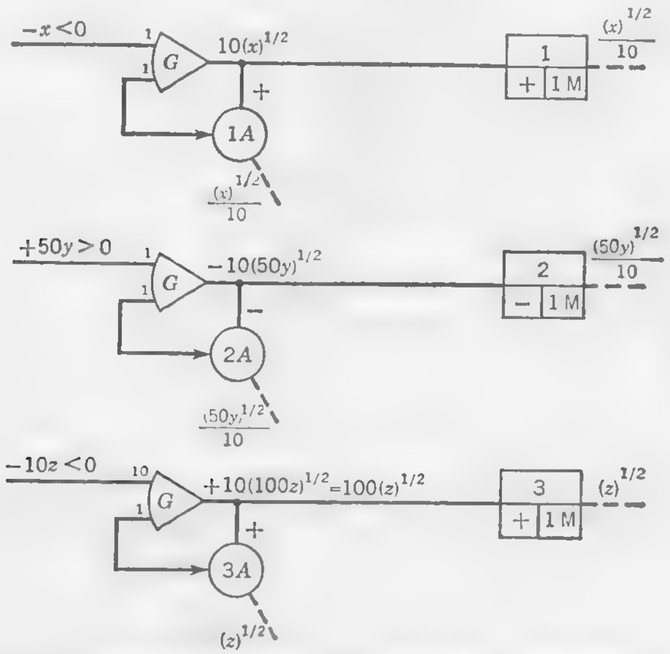


FIG. 5.1.9. Square-root extraction by implicit computation. Electronic Associates notation; note provisions for load-equalizing resistors.

(*d*) *Use of Electronic Multipliers.* Implicit-computation circuits analogous to Figs. 5.1.6 through 5.1.9 can be implemented by substitution of electronic-multiplier channels for servo potentiometer cups. It may be necessary to add small feedback capacitors (50 to 1,000$\mu\mu f$) to high-gain amplifiers to ensure feedback-loop stability and to filter electronic-multiplier ripple voltages (see also Sec. 3.2).

SCALING A PROBLEM

5.1.7. General. On the analog computer, it is necessary to ensure that voltages proportional to the problem variables, their derivatives, or any combination of these will never exceed the maximum allowable value of ± 100 volts. In addition, no voltage must change so rapidly that it will exceed the frequency limitations of the computing components or of the recording equipment. On the other hand, the voltages should not be so small that they are of the same order as the possible errors in a problem solution, and the problem should not take so long to solve that computer operation becomes tedious and inefficient.

Between these extremes there is a reasonable level for the voltage variables and a reasonable speed of solution. The task of determining these values for any particular problem involves the process of *scaling: amplitude scaling* to control the maximum level of the voltage variables, and *time scaling* to control the speed of solution.

5.1.8. Amplitude Scaling. Amplitude scaling translates the problem variables into voltages of the same order of magnitude. A distance s which varies from 0 to 10,000 ft, an angular acceleration $\ddot{\theta}$ which varies from -20 to 10 radians per sec^2, a temperature of 700°F, all these will be represented on the computer by voltages which

can vary from either 0 to 100 volts or -100 to $+100$ volts. For maximum accuracy, the voltages will be as large as possible without causing the equipment to overload. This implies that the maximum expected absolute value of the problem variable must be made to correspond to ± 100 volts. A *scale factor*, defined as the constant relating volts on the computer to one physical unit of the problem variable, can be introduced. This scale factor has dimensions of volts per physical unit, and its value can be determined by dividing 100 by the maximum expected absolute value of the problem variable. For ease of calculation, one increases the maximum expected absolute value to the nearest power of ten times 1, 2, or 5 before calculating the scale factor. Problem variables are then related to voltage by the equation

$$\text{Voltage variable} = \text{scale factor} \times \text{problem variable}$$

Table 5.1.1 demonstrates the determination of scale factors. Note the rounding off to the nearest power of 10 in the scale factor for T, and note that

1. Only powers of 10 times either *one*, *two*, or *five* are used. This is for ease in manipulating the equations as well as for convenience in interpreting the recorded results.
2. The dimensions are shown merely to remind one that all physical units have been converted to voltages, but this soon is taken for granted by the analog-computer user, and this column is generally not needed.
3. *If the derivative of a variable appears in a problem, then this must be scaled along with the variable; its scale factor need not be the same as that of the variable.*

Table 5.1.1. Sample Calculation of Scale Factors

Problem variable	Max absolute value	Scale factor	Dimensions
Linear distance s.....	10,000 ft	$\frac{100}{10{,}000} = \frac{1}{100}$	Volts per ft
Angular acceleration $\ddot{\theta}$.....	20 radians/sec^2	$\frac{100}{20} = 5$	Volts per radian per sec^2
Temperature T.....	700°F	$\frac{100}{700} > \frac{100}{1{,}000} = \frac{1}{10}$	Volts per °F

Having determined the amplitude scale factors, then the problem variables in the mathematical equations are replaced by the voltages representing them, adjustments being made to the coefficients throughout the equations in order to maintain equality. For example, s, $\ddot{\theta}$, and T would be replaced in the appropriate equation by $s/100$, $5\ddot{\theta}$, and $T/10$, and the related coefficients would be multiplied by 100, $\frac{1}{5}$, and 10. The problem equations are thus changed into voltage equations from which a computer circuit diagram can be drawn.

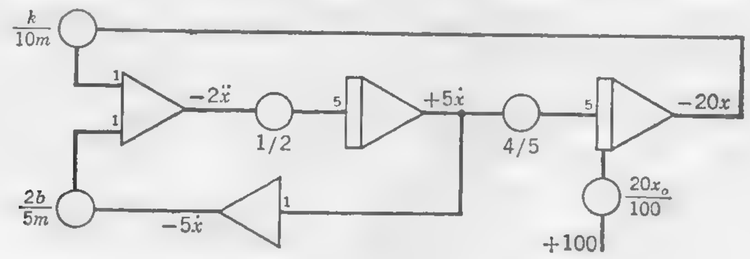
NOTE: *Resolver-servo inputs* may require a scale of $\frac{1}{4}$ volt per deg, $\frac{1}{2}$ volt per deg, 1 volt per deg, etc., depending on the design of the resolver and the range of the sine-cosine potentiometers used (see also Sec. 3.1).

5.1.9. Example: Solution of a Linear Differential Equation. To solve a differential equation solve each equation for the highest-order derivative and integrate successively to obtain the lower-order derivatives of the variable and the variable itself.

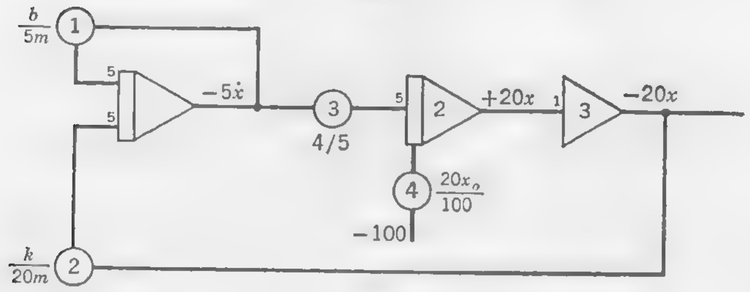
As an example consider the equation

$$m\ddot{x} + b\dot{x} + kx = 0$$

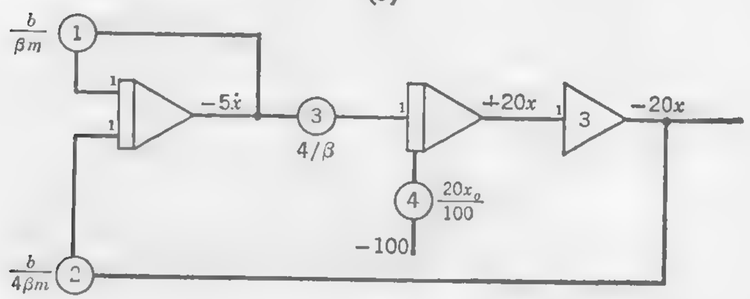
This is a second-order linear differential equation with constant coefficients. Such equations are important in studies of feedback control systems and the motions of



(*a*)



(*b*)



(*c*)

FIG. 5.1.10. Scaled block diagram for the linear differential equation $m\ddot{x} + b\dot{x} + kx = 0$ (*a*), simplified block diagram (*b*), and time-scaled block diagram with $\beta = 10$ (*c*).

bodies subject to spring and damping forces. Solving for the highest-order derivative yields

$$\ddot{x} = -\frac{b\dot{x}}{m} - \frac{k}{m}x \tag{5.1.1}$$

Scaling:

Variable	Max absolute value	Nearest power of 10 times 1, 2, or 5	Scale factor
x	5	5	$100/5 = 20$
$\dot{x}$	17	20	$100/20 = 5$
$\ddot{x}$	50	50	$100/50 = 2$

Constants and Initial Conditions:

$$k/m < 10 \qquad x_0 = +5$$
$$b/m < 1.5 \qquad \dot{x}_0 = 0$$

Voltage Equation. Multiplication and division of each variable by its scale factor yields the scaled voltage equation

$$[2\ddot{x}] = -\left(\frac{2b}{5m}\right)[5\dot{x}] - \left(\frac{2k}{20m}\right)[20x]$$

The bracketed terms represent computer voltages, and the terms in parentheses represent potentiometer settings. Figure 5.1.10*a* shows the scaled diagram.

An important alternative diagram exists if $\ddot{x}$ is not specifically required as an output. It then is no longer necessary to generate $\ddot{x}$ explicitly as the output of an amplifier, but rather the terms comprising $\ddot{x}$ can be fed directly to the inputs of an integrator, thereby yielding $\dot{x}$ as the output of this integrator. The scaling is slightly different, for though it is still necessary to solve for $\ddot{x}$, one writes the voltage equation *scaled at the level of* $\dot{x}$ as follows:

$$5\ddot{x} = -\left(\frac{5b}{5m}\right)[5\dot{x}] - \left(\frac{5}{20}\frac{k}{m}\right)[20x]$$

Since both potentiometer settings are greater than unity, it will be necessary to divide by 5 (or 10, if desired), and compensate for the gain reduction in the potentiometer by using input gains of 5 (or 10) to the $\dot{x}$ integrator. The final scaled equation is, therefore,

$$5\ddot{x} = -5\left(\frac{b}{5m}\right)[5\dot{x}] - 5\left(\frac{k}{20m}\right)[20x]$$

The constants not in parentheses are the integrator input gains. No bracket has been placed around $5\ddot{x}$, since it is not *explicitly generated*, thereby indicating that the last equation represents the input to an integrator which has $5\dot{x}$ as its output. The scaled diagram is shown in Fig. 5.1.10*b*.

5.1.10. Example: Solution of a Nonlinear Differential Equation. The Van der Pol oscillator is described by the differential equation

$$\ddot{x} - \lambda(1 - x^2)\dot{x} + x = 0$$

Scaling:

Variable	Max absolute value (to nearest power of 10 times 1, 2, or 5)	Voltage-scale factor
x	5	20
$\dot{x}$	10	10
$\ddot{x}$	20	5

Constants and Initial Conditions:

$$0.1 < \lambda < 1.0 \qquad x_0 = 1.0 \qquad \dot{x}_0 = 0$$

Solving for the highest-order derivative, and multiplying out the parentheses, we have

$$\ddot{x} = \lambda\dot{x} - \lambda x^2\dot{x} - x$$

Note that two multiplications will be required for the second term on the right-hand side. This will be accomplished with two cups on the x servo. The voltage equation is

$$[5\ddot{x}] = \left(\frac{5\lambda}{10}\right)[10\dot{x}] - \left(\frac{5\lambda}{4/10}\right)\left(\frac{20x}{100}\right)\left(\frac{20x}{100}\right)[10\dot{x}] - \left(\frac{5}{20}\right)[20x]$$

Since the potentiometer setting of the second term on the right-hand side is greater than 10, it will be necessary to divide it by 20 and compensate for the gain reduction in the potentiometer with an input gain of 20 to the $\ddot{x}$ amplifier. The final scaled equation is therefore

$$[5\ddot{x}] = \left(\frac{\lambda}{2}\right)[10\dot{x}] - 20\left(\frac{5\lambda}{8}\right)\left(\frac{20x}{100}\right)\left(\frac{20x}{100}\right)[10\dot{x}] - \left(\frac{1}{4}\right)[20x]$$

The scaled diagram is shown in Fig. 5.1.11.

5.1.11. Other Scaling Methods. For most problems, the amplitude-scaling procedures described in the previous paragraphs are the easiest and most efficient to use. Sometimes, however, it may be desirable to resort to other scaling procedures because

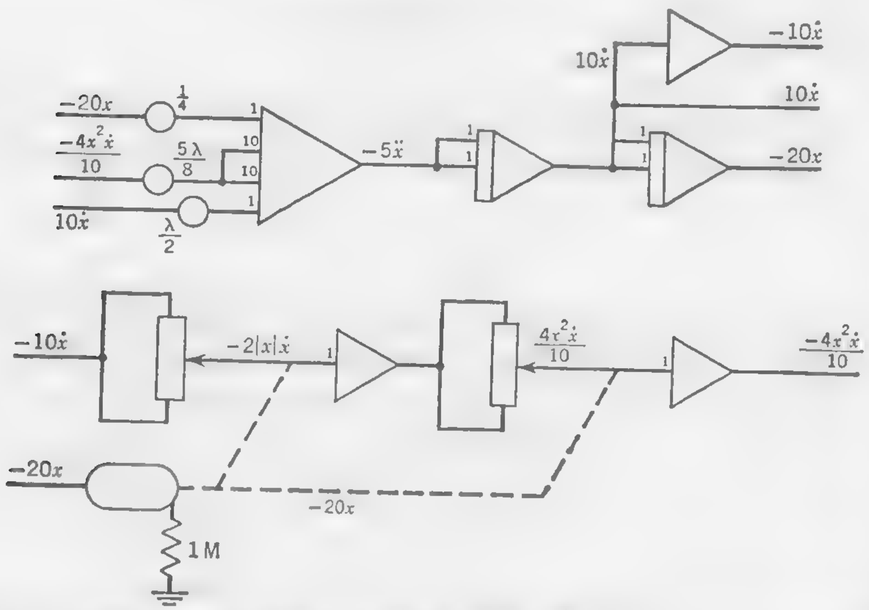


FIG. 5.1.11. Scaled block diagram for Van der Pol's differential equation. An isolation amplifier separates the two servo-driven potentiometers. Note how generation of $|x/5|^2$ instead of $(x/5)^2$ saves one phase inverter.

of the special nature of a problem. However, it can be shown that these other scaling procedures are merely extensions or special cases of the recommended procedures. For example, the use of transformation equations, which is another scaling procedure, merely involves an explicit definition or substitution of a computer variable X which can range from −100 to +100 volts (−1 to 1 *machine unit*) for the original variable x.

Sometimes an entire class of problems can be described by exactly the same set of equations, but the amplitudes of the variables change from one problem investigation to another. Such situations are handled conveniently with a slight modification of the recommended scaling procedure. This would merely require the substitution of general scale factors S_1, S_2, and S_3 instead of specific scale factors, such as 2, 5, 1/10, in the scaled equations. For example, referring to Eq. (1) again, we could have a general scaling as follows:

Generalized Scaling Variable	*Scale Factor*
x	S_1
$\dot{x}$	S_2
$\ddot{x}$	S_3

The scale factors S chosen for any particular problem would, of course, depend on the maximum absolute values expected for each variable in the particular problem.

The generalized scaled equation for Eq. (1) would then be

$$[S_3\ddot{x}] = -\left(\frac{S_3 b}{S_2 m}\right)[S_2\dot{x}] - \left(\frac{S_3 k}{S_1 m}\right)[S_1 x]$$

It is seen that the potentiometer settings would be functions of the scale factors S. To change from one scaled problem to the next one would merely have to choose the new scale factors S and recalculate the potentiometer settings.

5.1.12. Time Scaling. The ideal duration for a single dynamic solution obtained on most "slow" electronic analog computers is 10 to 60 sec. The minimum time is set by the performance of the computing units, particularly the servos and the output equipment, such as the xy plotters. The maximum time is essentially determined by convenience in the use of the computer. If a physical problem has dynamic solutions which last for much longer than 1 min or, alternatively, which contain natural frequencies too high for the computing and/or recording units, then a *time-scale change* will be necessary.

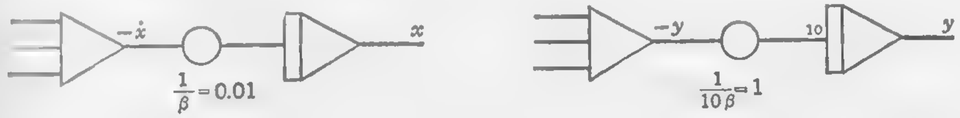
Experience has shown that the easiest method of time scaling consists of simply changing the gains of all integrators by the same constant factor, *increasing* the gain to decrease the time of solution, and vice versa. This change will *not* interfere with or alter any amplitude scaling.

The time-scale change is usually defined by the equation

$$T = \beta t$$

where T = machine time (real time)
t = original problem time
β = *time-scale factor* and has the units of machine time/original problem time

In order to slow down a problem (i.e., to cause machine time to be *larger* than original problem time), β is made greater than unity, and to speed up a problem (i.e., to cause machine time to be *smaller* than original problem time) β is made less than unity.



(*a*) (*b*)

FIG. 5.1.12. Typical integration circuits with $\beta = 100$ (*a*) and $\beta = 0.1$ (*b*).

The time-scale change is indicated on a computer diagram by *dividing* all inputs to integrators by the factor β. For example, if the maximum value of t is 0.4 sec, then it is convenient to let $\beta = 100$, so that the maximum machine time T will be 40 sec. A typical integration circuit is shown in Fig. 5.1.12*a*.

If, on the other hand, the maximum time t is given as 400 sec, then it is convenient to let $\beta = 0.1$, so that the maximum machine time T will be 40 sec (Fig. 5.1.12*b*).

If there is more than one input to an integrator, such as amplifier 1 in Fig. 5.1.10*b*, then both inputs must each be divided by the time-scale factor β. If the circuit of Fig. 5.1.10*b* were time-scaled, one would expect β to be either 5 to 10 (i.e., a slowdown). The settings of potentiometers 1, 2, and 3 would be divided by β. The setting of the initial-condition potentiometer 4 is *unchanged* (Fig. 5.1.10*c*).

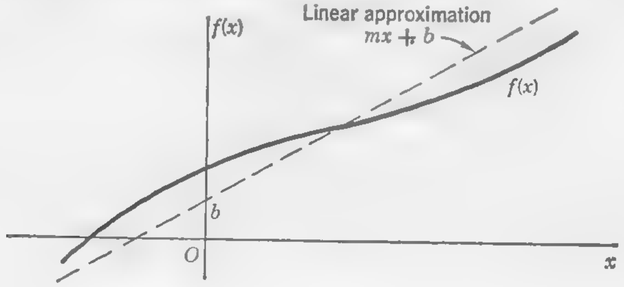
One final word on time scaling concerns *passive-element networks* which are often used to simulate certain transfer functions. If the problem is time-scaled, then the time constant of these passive-element networks must be changed by the same time-scale factor β, decreasing the time constant to decrease the time of solution ($\beta < 1$)

and increasing the time constant to increase the time of solution ($\beta > 1$). This means, in effect, that the time constants of the passive-element networks must be *multiplied* by β.

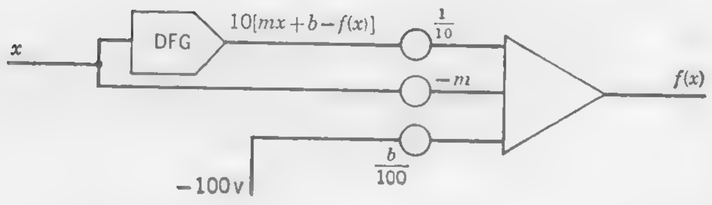
USE OF FUNCTION GENERATORS

5.1.13. General Rules. The following rules will minimize errors due to arbitrary-function-generating equipment (tapped potentiometers, curve followers, diode function generators, and cathode-ray function generators, Secs. 3.1 and 3.3):

1. Function generators should supply only *corrections* to analytic (most frequently linear) approximations to the desired function.
2. Every attempt should be made to have function-generator input and output voltages utilize the full dynamic range available.



(*a*)



(*b*)

FIG. 5.1.13. A typical function $f(x)$ (*a*), and generation of $f(x)$ as the sum of a linear approximation $mx + b$ and a small correction function (*b*). All function-generator errors are divided by 10.

To illustrate these rules, Fig. 5.1.13 shows how a typical function is generated as the sum of the linear approximation $mx + b$ and a correction function; *note how the function-generator output, and hence the function-generator error, is divided by a factor of* 10. Many diode function generators comprise internal circuits for generating $mx + b$ with accurate linear network components, so that the diode circuits generate only the correction function; correction function and diode-circuit errors are attenuated internally (see also Sec. 3.3).

5.1.14. Nonlinear Analytic Approximations. If accurate multipliers are available, function generators can be used to generate corrections to second- or third-order polynomial approximations.

A special scaling problem is presented by functions of the form $f(x) = Ae^{b/x}$, which occur frequently in chemical-reaction problems (Sec. 5.7); note that $Ae^{b/x}$ is often small over most of the range of x but increases rapidly for small x. Such functions are sometimes generated in the form

$$f(x) = \frac{g(x) + c}{(x + a)^2} \text{ with } g(0) = g(100) = 0$$

where $g(x) = Ae^{b/x}(x + a)^2 - c$ is a bell-shaped function of relatively small dynamic range, which is easily set up on a tapped-potentiometer function generator ("bell technique"). If the exponent can become very large, one may use the alternative form

$$f(x) = \frac{g(x)}{(a - x)^2} \qquad g'(100) = 0$$

so that $g(x)$ becomes a "semibell"-shaped function. Reference 1 discusses this and related techniques in detail.

SETUP SHEETS AND CHECKING PROCEDURES

5.1.15. Setup Sheets. In large analog computations, the computer is not set up directly by reference to the scaled block diagram but with the aid of *setup sheets*. These are form sheets prepared for each type of computing element and list the connections, input variables, and output variables of each computing element and recorder channel, as well as potentiometer settings and switch positions. A *prepatch check sheet* may remind the operator to connect integrator feedback networks, servo follow-up excitations, etc., which may not be directly indicated on the block diagram.

5.1.16. Checking Procedures. The purpose of problem checks is twofold. One objective is to verify the transcription of a set of equations to an analog-computer setup, and the other is to check the actual operation of each of the analog-computer components used in the simulation of a given physical system (see also Secs. 4.3 and 4.7). Actually, problem checking begins with the analysis of the equations which describe the physical system, to make sure that they are correct in themselves and that no additional information has been left out. Thereafter, the computer block diagram is checked against the equations of the system, and all setup sheets derived from the computer diagram should be double-checked against the original block diagram for transcription errors. Finally, the actual patching of the problem board is checked. Then one is ready to go on the computer, set the potentiometers, switches, functions, etc., and then perform the "problem check" which verifies that the computer setup simulates the system equations and that the equipment is functioning properly (Sec. 4.3).

The reason for this seemingly endless checking is to catch the errors as early in the problem setup as possible and thus to reduce the paper work required to correct the error. In practice, an error committed at the beginning of a problem setup, which is allowed to pass undetected to the final check, would probably generate other errors, and might seriously delay the start of the program. The use of the final "problem check" as the *only* check of a computer setup should be resorted to only in an emergency.

A typical setup and checking procedure for an analog-computer program is illustrated by the flow diagram of Fig. 5.1.14. Of all the checks shown, the "problem check" is the one that is least understood. A "problem check," to be valid, should check every point used in the computer. The simplest way to achieve this is to insert a compatible set of initial conditions on all integrators in the system, thus forcing every variable to have a given value at $t = 0$. It is usual practice to insert conditions for the variables so that they are 1, 2, or 5 times a power of 10, for ease in computing the remaining variables from the original equations. For example, one could go down the column of the amplifier sheets labeled "outputs" and insert the "problem check" values that should be read on the basis of the initial conditions chosen. The same can be done with servo sheets and resolver sheets. The outputs of the individual coefficient potentiometers need not be checked, since they are implicitly checked in reading the outputs or summing points of the amplifiers to which they lead. These can, if desired, be individually checked.

It is recommended that a calculation similar to the "problem check" calculation be performed for the actual initial conditions for the first run, thus checking that the computer has been restored to the correct initial conditions.

Now all that is required is to put the computer into OPERATE to obtain the first actual run. If all the previous checks have been performed about the only failures that can occur now to produce an erroneous result would be a relay failure on an integrator, or sluggish response of a servo. These failures can easily be detected and eliminated since most modern analog computers are equipped with special circuits for

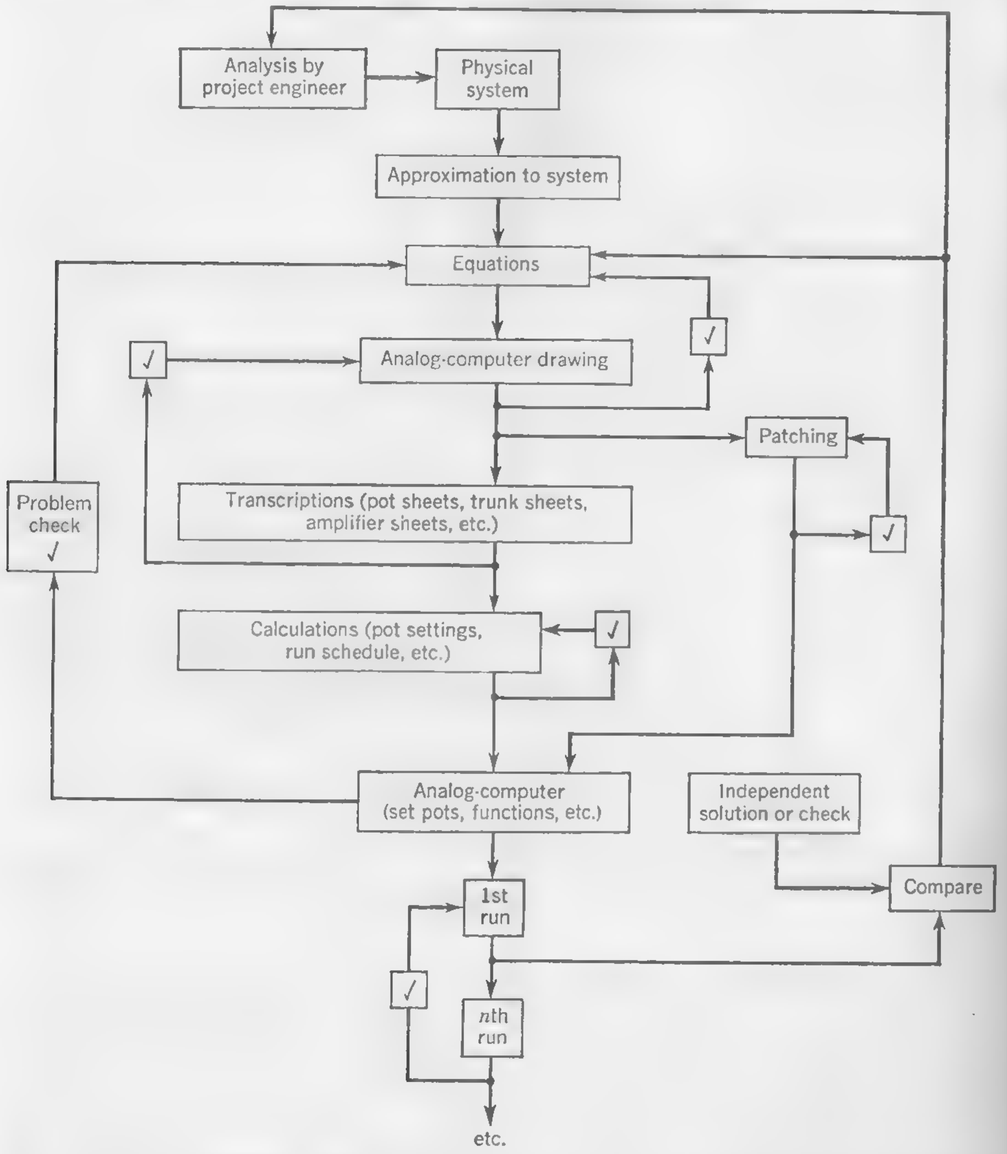


FIG. 5.1.14. A flow diagram showing the progress of an analog-computer problem from its inception to its final running. The squares with check marks indicate typical checks which should be performed, and result in "feedback" from one stage of the setup to a preceding stage.

checking relays associated with integrators and for quickly changing the speed of solution of the problem, thereby determining the effect of servo dynamics on the problem solution (Sec. 4.3).

5.1.17. Records and Summary of Problem-solving Procedure. A businesslike set of records summarizing the problem-solving procedure might include the following information.

1. *Statement of the problem in the original form*
   - *a.* Description of physical system and origin of problem
   - *b.* Equations and data describing the problem (this should be in the original form)
   - *c.* General objectives of study
   - *d.* Specific objectives, such as output data requirements
   - *e.* Expected first solution for computer check-out
2. *Preliminary problem investigation*
   - *a.* Data flow and functional block diagram
   - *b.* Approximations and simplifications
   - *c.* Special circuits and techniques
   - *d.* Recording techniques
   - *e.* Preliminary computer block diagram, equipment count
3. *Analog-computer equations*
   - *a.* Required calculations and listing of maximum values of important variables
   - *b.* Analysis, calculations, and other factors required to arrive at a completely scaled set of analog-computer equations
4. *Analog-computer block diagram and setup sheets*
   - *a.* Circuit diagrams
   - *b.* Potentiometer sheets
   - *c.* Multiplier sheets
   - *d.* Amplifier sheets
   - *e.* Recorder sheets
   - *f.* Function-generator sheets
   - *g.* Trunks
   - *h.* External connections
   - *i.* Special circuit descriptions (Purpose—approximations, setup, procedures)
   - *j.* Instructions to technicians
5. *Check procedure and first run*
   - *a.* Complete "problem check" (static check)
   - *b.* Calculations relevant to check
   - *c.* Special dynamic checks
   - *d.* Subsystem dynamic checks
   - *e.* Detailed conditions of first run
   - *f.* Possible solutions from other computers, such as digital solution
   - *g.* Check solution overlays (obtained from *f* or previous runs of same problem)
6. *Operating procedure*
   - *a.* Run schedule—potentiometer changes, function-generator changes, patching changes, switch positions, etc.
   - *b.* Recordings required
   - *c.* Other instructions or data needed in the operating procedure
7. *Running log*
   - *a.* Dates
   - *b.* Important times; such as starting and ending computation and transition points in the program
   - *c.* Changes and corrections to diagrams or patching
   - *d.* Malfunctions
   - *e.* Changes to problem unless covered in detail by the run schedule; parts of the run schedule may be included or expanded if appropriate
   - *f.* Any other information or thoughts which might contribute to better understanding of the problem at a later time
8. *Problem status at end of operating period*
   - *a.* Last solution for the record
   - *b.* List of potentiometer settings as read out at end of problem
   - *c.* List of switch positions and special conditions as observed at the end of problem
   - *d.* Revised copies of diagrams, potentiometer sheets, and other paper work of problem; original computer sheets should be revised so that they are up to date
9. *Review*
   - *a.* Name and address of sponsor—including department and group, if known

b. Names and titles of personnel—both sponsor and computer group associated with problem
c. General description of problem
d. Objectives
e. Brief history of operation and degree of success
f. Future plans

References

1. Rubin, A.: Function Generators, *Princeton Computing Center Rept.* 116, Electronic Associates, Inc., Princeton, N.J.

PART 2: MISCELLANEOUS APPLICATIONS

By VICTOR B. COREY

The examples of analog-computer applications presented in this section will serve to illustrate computer setup and scaling. All are relatively simple, but indicate the applicability of small analog computers to areas of physics, chemistry, and biology beyond the usual control-system applications. Several of the illustrative problems in this group are based on reference information in one or more of the following sources:

Sokolnikoff, I. S., and E. S. Sokolnikoff: *Higher Mathematics for Engineers and Physicists.*
Eldridge, J. A.: *The Physical Basis of Things.*
Glasstone, S.: *Principles of Nuclear Reactor Engineering.*
American Institute of Physics Handbook.
Bailey, N. T. J.: *The Mathematical Theory of Epidemics.*

All these problems have been adapted from *A Manual of Demonstration Problems for Analog Computers* by Victor B. Corey, to be published by Donner Scientific Company.

1. An automobile traveling 30 mph on a level straight road encounters an icy surface. The brakes are applied suddenly and the car skids. If the coefficient of friction between tires and icy roadway is 0.2, how long will it take to stop the car and how far will it go from the time and place the brakes were set?

$$\frac{d^2s}{dt^2} = -g\mu = -32.2(0.2) = -6.44 \text{ ft per sec}^2$$

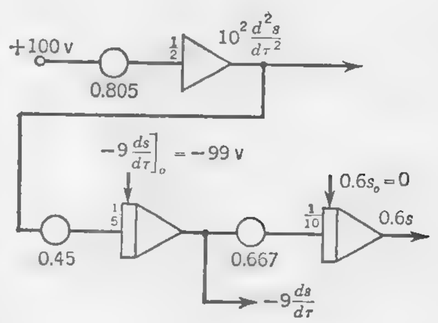
$v_0 = 44$ ft per sec

Change unit of time from 1 to 0.25 sec so that $t = \frac{1}{4}\tau$; $ds/dt = 4ds/d\tau$; $d^2s/dt^2 = 16d^2s/d\tau^2$:

$$\frac{d^2s}{d\tau^2} = -0.4025\left.\frac{ds}{d\tau}\right]_0 = \left.\frac{1}{4}\frac{ds}{dt}\right]_0 = 11 \text{ ft per unit of } \tau$$

Solve on computer with τ in seconds so 1 sec of solution represents 0.25 sec in physical problem. Let s, $ds/d\tau$, and $d^2s/d\tau^2$ be represented by computer voltages of $0.6s$, $9ds/d\tau$, and $2 \times 10^2\, d^2s/d\tau^2$, respectively.

$$\left(2 \times 10^2 \frac{d^2s}{d\tau^2}\right) = -2 \times 10^2(0.4025) = -80.5$$



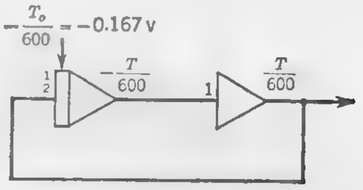
2. What is the limiting constraining force T that a sailor can develop on a ship at dock if the hawser from the bow is wrapped two full turns around a nonrotating capstan, and the sailor pulls on the free end with a force of 100 lb? Assume coefficient of friction $\mu = 0.5$.

$$\frac{dT}{d\theta} = \mu T = 0.5T \qquad T_0 = 100 \text{ lb}$$

Solve the following analogous equation on the computer with τ in seconds so 1 sec of time in computer solution corresponds to 1 radian of angle in physical problem. Let $T/600$ be the number of volts representing a tension of T lb. Equation for computer is then

$$\left(\frac{1}{600}\frac{dT}{d\tau}\right) = 0.5\left(\frac{T}{600}\right)$$

$$\frac{T_0}{600} = 0.167 \text{ volt}$$



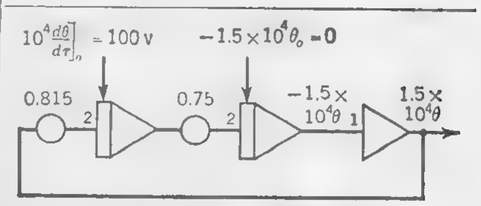
3. A small plumb bob is suspended from a fixed point by a string 100 cm long. Describe the time behavior of this simple pendulum if the mass, initially at rest in its equilibrium position, is suddenly given a tangential velocity of 2 cm per sec.

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\theta = -9.8\theta$$

$$\left.\frac{d\theta}{dt}\right]_0 = 0.02 \text{ radian per sec}$$

Change unit of time from 1 to 0.5 sec so that $t = 0.5\tau$; $d\theta/dt = 2d\theta/d\tau$; $d^2\theta/dt^2 = 4d^2\theta/d\tau^2$:

$$\frac{d^2\theta}{d\tau^2} = -2.45\theta\left.\frac{d\theta}{d\tau}\right]_0 = 10^{-2} \text{ radian per unit of } \tau$$

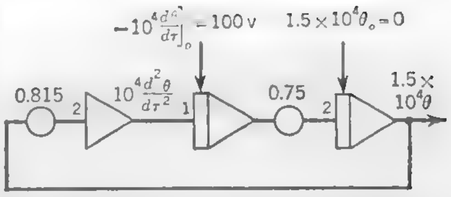


$d^2\theta/d\tau^2$ does not appear explicitly and therefore needs no scaling consideration in this arrangement.

Based on estimates of 6.4×10^{-3} radian, 10^{-2} radian per unit of τ, and 1.56×10^{-2} radian per (unit of τ)^2 for maximum values of θ, $d\theta/d\tau$, and $d^2\theta/d\tau^2$, respectively, let these variables be represented on the computer by voltages $1.5 \times 10^4\theta$, $10^4\, d\theta/d\tau$, and $6 \times 10^3\, d^2\theta/d\tau^2$, respectively. The computer equation becomes

$$\left(10^4 \frac{d^2\theta}{d\tau^2}\right) = -0.815(2)(1.5 \times 10^4\theta)$$

$$\left.10^4 \frac{d\theta}{d\tau}\right]_0 = 100 \text{ volts}$$



This arrangement is also logically correct, but $d^2\theta/d\tau^2$ appears explicitly, and its scaling must be changed to avoid overload of the first amplifier.

4. Water flows out through a 1-in.-diameter circular orifice in the side of a vertical cylindrical tank 2 ft in diameter. The velocity of water in the jet is $\sqrt{2gh}$ where h is height of the water surface in the tank above the center of the orifice. How long will it take the water level to fall from 25 to 9 ft above the orifice?

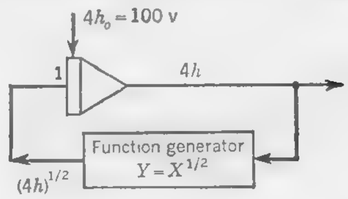
$$\frac{dh}{d} = -\left(\frac{d}{D}\right)^2 \times (2gh)^{1/2} = -\frac{1}{72}h^{1/2} \qquad h_0 = 25 \text{ ft}$$

Change unit of time from 1 to 36 sec so $t = 36\tau$:

$$\frac{dh}{d\tau} = -\frac{1}{2}h^{1/2}$$

Solve on computer with τ in seconds so 1 sec in solution represents 36 sec in problem. Let $4h$ be computer voltage representing h.

$$4\frac{dh}{d\tau} = -(4h)^{1/2} \qquad 4h_0 = 100 \text{ volts}$$



5. An average shallow-shock earthquake has an observed frequency of 3 cps. Initial energy in the quake is 70×10^{23} ergs. If the mechanical Q of the rock substratum which transmits this disturbance is 100, how long will be required for the energy in the quake to be reduced by a factor of 100?

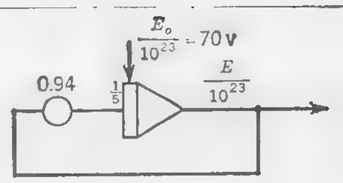
$$\frac{\Delta E/\text{cycle}}{E} = -\frac{2\pi}{Q} = -2\pi 10^{-2} = -\frac{1}{E}\frac{dE}{dt}\frac{1}{\text{frequency}}$$

$$\frac{dE}{dt} = -6.28 \times 10^{-2} \times 3E = -0.188E$$

$E_0 = 70 \times 10^{23}$ ergs

Represent problem time by computer time τ directly. Let computer voltage of $E/10^{23}$ represent E. Then computer equation becomes

$$\left(\frac{1}{10^{23}}\frac{dE}{d\tau}\right) = -0.188 \times \left(\frac{E}{10^{23}}\right) \qquad \frac{E_0}{10^{23}} = 70 \text{ volts}$$



6. A horizontal bar of 2S aluminum has a rectangular cross section of 1 in. height and 0.25 in. width. One end of the bar is built in, supporting a free length of 3 ft under its own weight. How far does the bar deflect as a function of length from the built-in end? What is the inclination of the bar to the horizontal at its free end? Density of the bar is 169 lb per cu ft and Young's modulus is 10^7 psi.

$$\frac{d^2y}{dx^2} = -\frac{M}{EI} = -\frac{1}{2}\frac{g\rho}{EI}(l^2 - x^2)$$

$$I = \frac{2}{3}a^3b = \frac{1}{48} \qquad g\rho = \frac{1}{4} \times \frac{169}{1{,}728}\ \text{psi} \qquad l = 36\ \text{in.}$$

$$\frac{d^2y}{dx^2} = -5.9 \times 10^{-8}(1.295 \times 10^3 - x^2) \qquad y = 0 \text{ at } x = 0$$

$$\frac{dy}{dx} = 0 \text{ at } x = 0$$

Solve the following analogous equation on the computer with τ in seconds of computer time so 1 sec in computer solution corresponds to 1 in. in the physical problem:

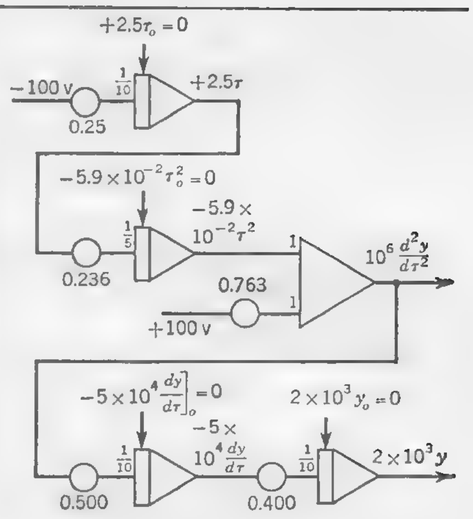
$$\frac{d^2y}{d\tau^2} = -5.9 \times 10^{-8}(1.295 \times 10^3 - \tau^2) \qquad y = 0 \text{ at } \tau = 0$$

$$\frac{dy}{p} = 0 \text{ at } \tau = 0$$

Let computer voltages of $2 \times 10^3 y$, $5 \times 10^4\, dy/d\tau$, and $10^6\, d^2y/d\tau^2$ represent y, $dy/d\tau$, and $d^2y/d\tau^2$, respectively. The equation arranged for computer solution becomes

$$\left(10^6 \frac{d^2y}{d\tau^2}\right) = -5.9 \times 12.95 + 5.9 \times 10^{-2}\tau^2$$

$$= -100(0.763) + 5.9 \times 10^{-2}\tau^2$$



7. An insulated reservoir is operated at its capacity of 160 lb weight. Incoming fluid at temperature θ is furnished to the reservoir at a constant rate of 5 lb per min. After thorough mixing to achieve a uniform temperature θ in the reservoir the fluid leaves at a constant rate of 5 lb per min. The system has been operating in equilibrium at 20°C throughout. If the temperature of the inlet water is changed suddenly to 85°C and maintained there, how long will be required for the outlet water to reach 80°C? (Neglect heat capacity of reservoir structure.)

(Mass flow per min) × (specific heat of fluid) × $(\theta_1 - \theta)$ = (mass of fluid in reservoir) × (specific heat of fluid) $\frac{d\theta}{dt}$

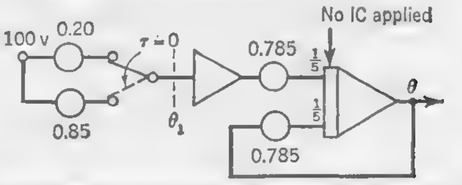
$$mC(\theta_1 - \theta) = MC\frac{d\theta}{dt}$$

$$\frac{d\theta}{dt} = -\frac{m}{M}(\theta - \theta_1) = -\frac{5}{160}(\theta - \theta_1) = -0.0313(\theta - \theta_1)$$

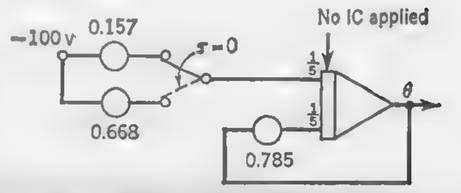
Change unit of time from 1 to 5 min so $t = 5\tau$; $d\theta/dt = \frac{1}{5}\, d\theta/d\tau$:

$$\frac{d\theta}{d\tau} = -0.157(\theta - \theta_1)$$

Solve equation of the same form on the computer with τ in seconds. Then 1 sec in the computer solution represents 5 min in the physical problem. Let θ and θ_1 be represented, respectively, by numerically equal voltages.



Arrangement permitting convenient change of parameters.



Arrangement requiring only one amplifier.

8. When used in liquids a thermometer is found to have a time constant of 0.2 sec. If the thermometer is removed from temperature equilibrium in ice and water at 0°C and is immersed suddenly in boiling water at 100°C, how long after immersion will the temperature reading be within 1°C of the boiling-water bath?

$$\frac{d\theta}{dt} = -\frac{1}{0.2}(\theta - \theta_b) = -5(\theta - \theta_b)$$

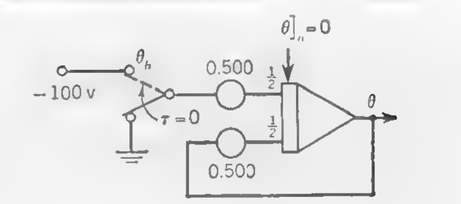
where θ = temperature reading of thermometer, °C
θ_b = temperature of bath, °C

Change unit of time from 1 to $\frac{1}{20}$ sec so $t = \frac{1}{20}\tau$; $d\theta/dt = 20\, d\theta/d\tau$:

$$\frac{d\theta}{d\tau} = -0.25(\theta - \theta_b) \qquad \theta]_0 = 0$$

$$\theta_b = 0 \qquad \tau < 0$$
$$\theta_b = 100 \qquad \tau \geq 0$$

Let θ and θ_b be represented, respectively, on the computer with numerically equal voltages.



9. The temperature inside an icebox is maintained at −20°C. Insulation is provided by a layer of homogeneous material 8 cm in thickness, contained between inside and outside metal walls. Conductivity of the insulating material is 0.0002 cal per sec per cm² per °C per cm. How many calories will be conducted through the wall area of 9,500 cm² during a 10-hr period if the outside temperature rises at a constant rate from 20 to 35°C over this interval?

Rate of heat conduction = conductivity × area × temperature gradient

$$\frac{dQ}{dt} = kA\frac{d\theta}{dx} = \frac{kA(T_{\text{outside}} - T_{\text{inside}})}{\text{wall thickness}}$$

$$= \frac{2 \times 10^{-4} \times 9.5 \times 10^3}{8}[20 + 4.16 \times 10^4 t - (-20)]$$

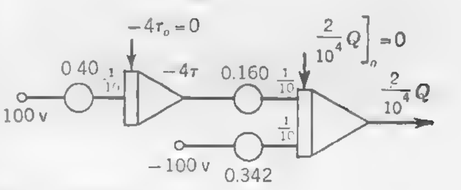
$$= 9.5 + 9.9 \times 10^{-5}t$$

Change unit of time from 1 sec to $\frac{1}{2}$ hr so that $t = 1{,}800\tau$; $dQ/dt = 1/1{,}800\, dQ/d\tau$:

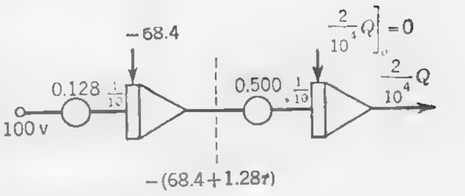
$$\frac{dQ}{d\tau} = 1.71 \times 10^4 + 320\tau$$

Let $2Q/10^4$ be the computer voltage representing Q. Solve the analogous equation with τ as computer time in seconds, so that 1 sec in the computer solution corresponds to $\frac{1}{2}$ hr in the physical problem:

$$\left(\frac{2}{10^4}\frac{dQ}{d\tau}\right) = 3.42 + 0.064\tau = \frac{1}{20}(68.4 + 1.28\tau)$$



First alternative



Second alternative

10. Neglecting effects of temperature, the pressure (as well as the density) of the earth's atmosphere decreases with altitude at a rate proportional to atmospheric pressure (or density) itself. Pressure of the atmosphere is in fact reduced by half for each additional 3.5 miles of altitude. What is the pressure at an altitude of 50,000 ft above the ground where atmospheric pressure is 740 mm of mercury?

$$\frac{dp}{dh} = -\frac{0.693}{3.5 \times 5{,}280}\,p = -3.76 \times 10^{-5}p$$

Change unit of height from 1 to 2,000 ft so that $h = 2{,}000\,H$; $dp/dh = 1/2{,}000\;dp/dH$:

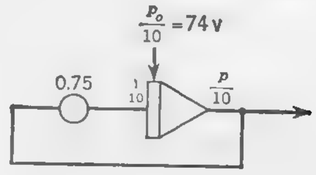
$$\frac{dp}{dH} = -0.075 \qquad p = p_0 = 740 \text{ mm Hg when } H = 0$$

Set up and solve the analogous equation

$$\frac{dp}{d\tau} = -0.075$$

where units of 1 sec in the computer solution correspond with units of 2,000 ft in the physical problem. Let $p/10$ be the computer voltage representing p. The computer equation is

$$\left(\frac{1}{10}\frac{dp}{d\tau}\right) = -0.075\left(\frac{p}{10}\right) \qquad \frac{p_0}{10} = 74 \text{ volts}$$



11. The mass-absorption coefficient (linear-absorption coefficient divided by density) for gamma rays of 1 Mev energy is approximately 0.068 per cm for lead (density 11.3 g per cm³) and 0.067 per cm for water. If a beam of 1-Mev gamma radiation is attenuated first by 35 cm of water and then by a lead shield 3 cm thick, what percentage of the initial intensity will emerge?

$$\frac{dI}{dx} = -\left(\frac{\mu}{\rho}\right)\rho I$$

For water: $\dfrac{dI}{dx} = -0.067I \qquad 0 \le x \le 35$ cm

$$I = I_0 \text{ at } x = 0$$

For lead: $\dfrac{dI}{dx} = -0.068(11.3)I$

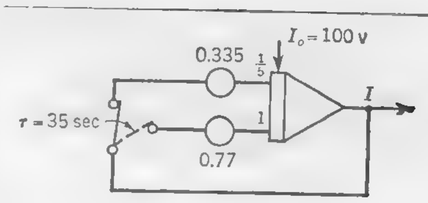
$$= -0.77I \qquad 35 \le x \le 38 \text{ cm}$$

Solve analogous equations of same form on computer:

$$\frac{dI}{d\tau} = -0.067I \qquad \begin{cases} 0 \le \tau \le 35 \text{ sec} \\ I = I_0 = 100 \text{ volts at } \tau = 0 \end{cases}$$

$$\frac{dI}{d\tau} = -0.77I \qquad 35 \le \tau \le 38 \text{ sec}$$

where τ is in seconds. Find I at 38 sec. (Note that different time scaling could be used for the two portions of the absorption path.)



12. After operating for some time at constant power level, a thermal-neutron flux level a reactor is shut down abruptly. Iodine 135 and xenon 135 are then present in amounts of 7.25×10^{16} and 3.05×10^{16} atoms per unit volume, respectively. The half lives of these two nucleides are 6.7 and 9.2 hr, respectively. I 135 decays to Xe 135, which in turn decays to comparatively stable cesium 135. Find concentrations of I 135 and Xe 135 as functions of time after shutdown.

$$\frac{dI}{dt} = -\frac{0.693}{6.7}\,I = -0.104I$$

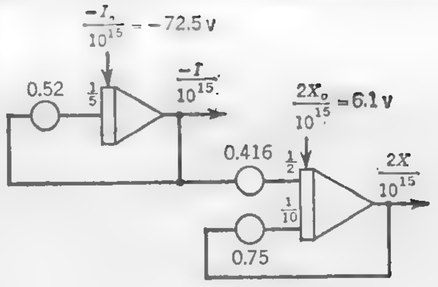
$$\frac{dX}{dt} = -\frac{0.693}{9.2}\,X + 0.104I = -0.075X + 0.104I$$

Solve analogous equations on computer with τ in seconds. Then 1 sec in computer solution corresponds to 1 hr in physical problem. Let $I/10^{15}$ be computer voltage to represent I. Let $2X/10^{15}$ be computer voltage to represent X. The computer equations become

$$\left(\frac{1}{10^{15}}\frac{dI}{d\tau}\right) = -0.104\left(\frac{I}{10^{15}}\right) \qquad \frac{I_0}{10^{15}} = 72.5 \text{ volts}$$

$$\left(\frac{2}{10^{15}}\frac{dX}{d\tau}\right) = -0.075\left(\frac{2X}{10^{15}}\right) + 0.208\left(\frac{I}{10^{15}}\right)$$

$$\frac{2X_0}{10^{15}} = 6.1 \text{ volts}$$



13. A foil of indium 115 weighing 0.050 g is irradiated for exactly 1 hr in a thermal-neutron flux of 4×10^9 neutrons per cm² per sec. Indium 116 is produced through a neutron-gamma reaction with cross section $\sigma = 190$ barns (1 barn = 10^{-24} cm²). The indium 116 is a radioactive beta emitter with half life of $T_{1/2} = 54$ min. It has a negligible cross section for thermal-neutron capture. What will be the activity of the foil (disintegrations per sec) just 3 hr after it is removed from the thermal-neutron flux? Avogadro's number is $N_a = 6.02 \times 10^{23}$ atoms per gram-atomic weight.

Let X = number of indium 115 atoms in foil
Let Y = number of indium 116 atoms in foil

$$X_0 = \frac{wN_a}{W} = \frac{0.050 \times 6.02 \times 10^{23}}{115} = 2.62 \times 10^{20} \text{ atoms}$$

$$\frac{dX}{dt} = -X\sigma_x\phi \text{ so that } \left.\frac{dX}{dt}\right]_0 = -X_0\sigma_x\phi$$

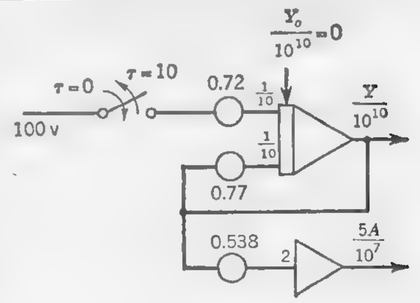
$$= -2.62 \times 10^{20} \times 190 \times 10^{-24} \times 4 \times 10^9$$
$$= -1.99 \times 10^8 \text{ atoms per sec}$$

But at this rate X is not reduced significantly from its initial value in 1 hr (or 1 year!) of irradiation, so that $X\sigma_x\phi \approx X_0\sigma_x\phi = 1.99 \times 10^8$ atoms per sec

$$\frac{dY}{dt} = +X\sigma_x\phi - Y\sigma_y\phi - \frac{0.693}{T_{1/2}}\,Y$$

But since $\sigma_y \approx 0$ and $X\sigma_x\phi \approx 1.99 \times 10^8$ atoms per sec

$$\frac{dY}{dt} = 1.99 \times 10^8 - \frac{0.693}{54 \times 60}\,Y \text{ during irradiation}$$



Change unit of time from 1 sec to 0.1 hr (6 min) so that $t = 360\tau$; $dY/dt = 1/360\, dY/d\tau$:

$$\frac{dY}{d\tau} = 7.2 \times 10^{10} - 0.077Y \qquad 0 \le \tau \le 10$$

$$Y = 0 \text{ at } \tau = 0$$

Activity of indium 116 is

$$A = (\lambda \text{ per sec})N = 2.15 \times 10^{-4}N$$

Solve the analogous equations on the computer with τ in seconds. Then 1 sec in computer solution corresponds to 0.1 hr = 6 min in physical problem. Let $Y/10^{10}$ and $5A/10^7$ be computer voltages representing Y and A, respectively. Computer equations become

$$\left(\frac{1}{10^{10}}\frac{dY}{d\tau}\right) = 7.2 - 0.077\left(\frac{Y}{10^{10}}\right) \qquad \left(\frac{Y_0}{10^{10}}\right) = 0$$

$$\left(\frac{5A}{10^7}\right) = 2(0.538)\left(\frac{Y}{10^{10}}\right)$$

14. A sample of polluted water contains three different groups of bacteria which show different degrees of resistance to the germicidal action of a disinfectant. Each of the separate groups shows an exponential survival characteristic under the treatment contemplated. During the first minute of treatment, 50% of the first group, 30% of the second, and 20% of the third are destroyed. If the population of each of the three groups is 10^5 per ml at the beginning of treatment, find the population in each group as a function of time. Find the total number of surviving bacteria vs. time and determine the period of treatment required to reduce total number of survivors to 1% of the original level.

5-30

Let x, y, and z represent surviving bacteria in the three groups, respectively.

$$\frac{x_0}{x_1} = \frac{100}{50} = 2 = e^{0.693} \qquad \therefore \lambda_x = 0.693 \text{ per min}$$

$$\frac{y_0}{y_1} = \frac{100}{70} = 1.43 = e^{0.358} \qquad \therefore \lambda_y = 0.358 \text{ per min}$$

$$\frac{z_0}{z_1} = \frac{100}{80} = 1.25 = e^{0.223} \qquad \therefore \lambda_z = 0.223 \text{ per min}$$

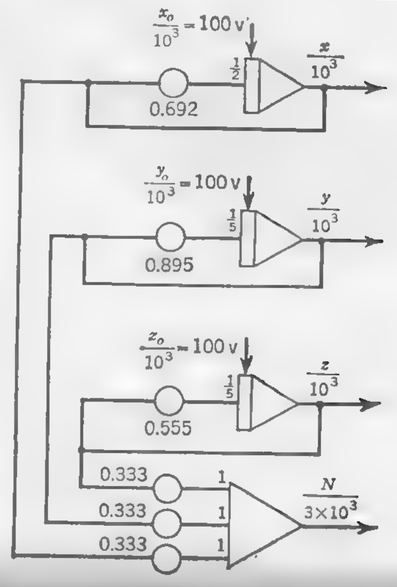
$$\frac{dx}{dt} = -0.693x \qquad \frac{dy}{dt} = -0.358y \qquad \frac{dz}{dt} = -0.223z$$

Change unit of time from 1 to ½ min so $t = \frac{1}{2}\tau$:

$$\frac{dx}{d\tau} = -0.346x \qquad \frac{dy}{d\tau} = -0.179y \qquad \frac{dz}{d\tau} = -0.111z$$

$$x_0 = y_0 = z_0 = 10^5 \text{ per ml at } \tau = 0$$

Solve analogous equations on computer with τ in seconds. Then 1 sec in computer solution corresponds to 30 sec in physical problem. Let $x/10^3$, $y/10^3$, $z/10^3$ be computer voltages representing x, y, z, respectively. Let $N = x + y + z$. Then $N_0 = 3 \times 10^5$ per ml. Let $N/(3 \times 10^3)$ be computer voltage representing N. The equations for the



$$\left(\frac{1}{10^3}\frac{dx}{d\tau}\right) = -0.346\left(\frac{x}{10^3}\right) \qquad \left(\frac{1}{10^3}\frac{dy}{d\tau}\right) = -0.179\left(\frac{y}{10^3}\right)$$

$$\left(\frac{1}{10^3}\frac{dz}{d\tau}\right) = -0.111\left(\frac{z}{10^3}\right)$$

$$\left(\frac{N}{3 \times 10^3}\right) = \frac{1}{3}\left(\frac{x}{10^3} + \frac{y}{10^3} + \frac{z}{10^3}\right)$$

$$\frac{x_0}{10^3} = \frac{y_0}{10^3} = \frac{z_0}{10^3} = 100 \text{ volts} \qquad \frac{N_0}{3 \times 10^3} = 100 \text{ volts}$$

15. A swinging door is returned toward its rest position by a nonlinear spring. Within ±20 deg of the rest position the spring is linear and generates a restoring torque of 18 lb-ft per radian of deflection. Beyond ±20 deg deflection the restoring torque from the spring increases with angle at only one-fourth the rate which applies near the rest position. The moment of inertia of the door about the hinge axis is 0.52 lb-ft² per g. Oscillation of the door is damped by a viscous torque of 0.2 lb-ft per radian per sec of angular velocity. A doorstop limits the outward opening of the door to +110 deg. (*a*) If a waiter kicks open the door with a maximum angular velocity of 90 deg per sec at +10 deg deflection, how far will the door open? (*b*) With what angular velocity must the door pass through its rest position in order just to reach the doorstop under spring and damping forces above? (*c*) If the door is released from rest in contact with the doorstop, how long will it take to reach the equilibrium position?

5-31

$$0.52\frac{d^2\theta}{dt^2} + 0.2\frac{d\theta}{dt} + T(\theta) = 0$$

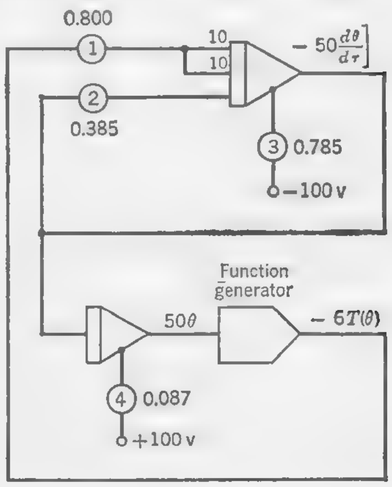
where θ is measured in radians. For small oscillations $T(\theta) = 18\theta$ and

$$f_n = \frac{1}{2\pi}\sqrt{\frac{18}{0.52}} \approx 0.94 \text{ cps}$$

Let computer time τ represent problem time directly. From the problem statement $T(\theta)$ has a maximum value less than 15. Let $6T(\theta)$ be the computer voltage to represent $T(\theta)$. Similarly, θ has a maximum value less than 2 radians. Let 50θ be the computer voltage to represent θ. $d\theta/dt$ has a value of 1.57 radians per sec when $\theta = \theta_0 = 0.174$ radian. Let $50\, d\theta/d\tau$ be the computer voltage to represent $d\theta/d\tau$. The equation for the computer becomes

$$-\left(50\frac{d^2\theta}{d\tau^2}\right) = 0.385\left(50\frac{d\theta}{d\tau}\right) + 16.0[6T(\theta)]$$

$$\left(50\frac{d\theta}{d\tau}\right)_0 = 78.5 \text{ volts} \qquad 50\theta_0 = 8.7 \text{ volts}$$



16. In a total population of 1,000 individuals there are 10 infectious carriers of a contagious disease. 900 of the population are susceptible to the disease, while the remaining 90 have recovered and are immune. Based on homogeneous mixing of the population, each infectious carrier on an average infects one per thousand of the remaining susceptible members every day while he can communicate the disease. After 14 days each infective recovers and becomes immune. Show as functions of time (*a*) the number x of susceptibles; (*b*) the number y of infectives; (*c*) the number z recovered and immune; and (*d*) the rate $-dx/dt$ at which new cases occur (the epidemic curve).

$$\frac{dx}{dt} = -\beta xy$$

$$\frac{dy}{dt} = \beta xy - \gamma y$$

$$\frac{dz}{dt} = \gamma y$$

If t = time in days, $\beta = 0.001$ and $\gamma = 1/4 = 0.072$. $\therefore$ the equations become

$$\frac{dx}{dt} = -0.001xy \qquad x_0 = 900$$

$$\frac{dy}{dt} = 0.001xy - 0.072y \qquad y_0 = 10$$

$$\frac{dz}{dt} = 0.072y \qquad z_0 = 90$$

Solve the following analogous equations on the computer where τ is computer time in seconds so that 1 sec in the computer solution corresponds to 1 day in the physical problem.

$$\frac{dx}{d\tau} = -0.001xy \qquad x_0 = 900$$

$$\frac{dy}{d\tau} = 0.001xy - 0.072y \qquad y_0 = 10$$

$$\frac{dz}{d\tau} = 0.072y \qquad z_0 = 90$$

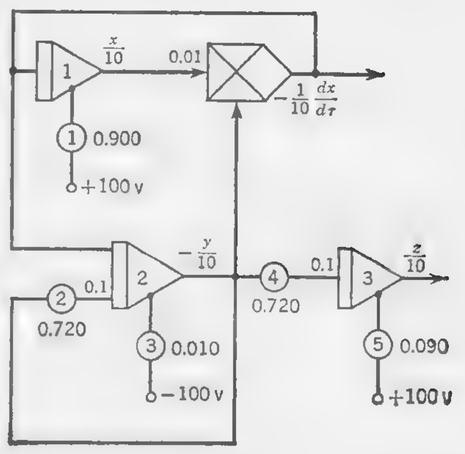
$x_{max} = 900$. Let $x/10$ be computer voltage to represent x. Estimate $y_{max} = 500$. Let $y/10$ be computer voltage to represent y. Estimate $z_{max} = 1{,}000$. Let $z/10$ be computer voltage to represent z. The equations for the computer become

$$-\frac{1}{10}\frac{dx}{d\tau} = \left(0.01\,\frac{x}{10}\,\frac{y}{10}\right)$$

$$\frac{1}{10}\frac{dy}{d\tau} = \left(0.01\,\frac{x}{10}\,\frac{y}{10}\right) - 0.072\,\frac{y}{10}$$

$$\frac{1}{10}\frac{dz}{d\tau} = \left(0.072\,\frac{y}{10}\right)$$

$\frac{x_0}{10}$ = 90 volts $\qquad \frac{y_0}{10}$ = 1.0 volts $\qquad \frac{z_0}{10}$ = 9 volts



17. In certain oscillatory systems governed by Mathieu's equation (useful in the study of frequency-modulation phenomena) the stiffness coefficient, i.e., the coefficient of the displacement variable itself, is a periodic function of time. Investigate the stability and periodicity of a system represented by the equatiou

$$\frac{d^2y}{dt^2} + (b - 14\cos 2t)y = 0$$

for values of b as follows: 20; 17; 15; 10; 9; 8; 7.

$$\frac{d^2y}{dt^2} + (20 - 14\cos 2t)y = 0$$

Let computer time τ represent problem time t directly. Let $y_{max} = 100$; $\therefore$ let y be represented by a numerically equal computer voltage.

$(20 - 14\cos 2\tau)$ has a maximum value of 34. Let the computer voltage to represent $(20 - 14\cos 2\tau)$ be $[2(20 - 14\cos 2\tau)]$.

$$\text{Estimated } \left[\frac{d^2y}{d\tau^2}\right]_{max} = 3{,}400 \text{ units of } y \text{ per sec}^2$$

Estimated $\left[\frac{dy}{d\tau}\right]_{max}$:

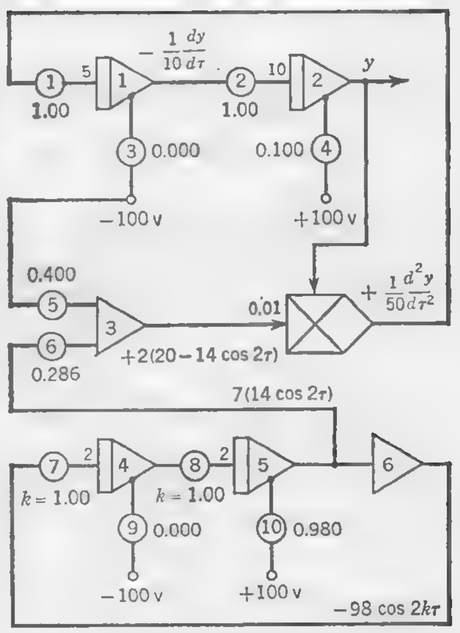
$$\frac{3{,}400}{\dot{y}_{max}} = \frac{\dot{y}_{max}}{100}$$

$$\therefore \text{ estimated } \left[\frac{dy}{d}\right]_{max} \approx 600 \text{ units of } y \text{ per sec}$$

Let $\frac{dy}{d\tau}$ be represented by computer voltage $\left(\frac{1}{10}\frac{dy}{d\tau}\right)$

The computer equation becomes

$$\frac{2}{100}\frac{d^2y}{d\tau^2} = -\frac{[2(20 - 14\cos 2\tau)]y}{100}$$



18. Based on statistics for the period since 1925 and UN predictions to the year 2000, world population is multiplying at an ever-increasing rate. In 1925 the world population of 1,907 million was increasing at a rate which would produce 23% population increase in 25 years. The rate of (exponential) multiplication is itself increasing almost linearly with time, such that in the year 2000 the world population will be increasing at a rate which would produce 64% population increase in 25 years. On these assumptions, find world population vs. time between 1925 and 2025.

At $t = 0$ (in 1925)

$$1.23 = e^{k25} = e^{0.206}$$

$$\therefore k_{t=0} = 8.3 \times 10^{-3} \text{ per year}$$

At $t = 75$ (in 2000)

$$1.64 = e^{k25} = e^{0.495}$$

$$\therefore k_{t=75} = 1.98 \times 10^{-2} \text{ per year}$$

$$\therefore k = 8.3 \times 10^{-3} + \left(\frac{19.8 - 8.3}{75}\right) 10^{-3}t$$

$$= 8.3 \times 10^{-3} + 1.53 \times 10^{-4}t$$

$$\frac{dN}{dt} = kN = (8.3 \times 10^{-3} + 1.53 \times 10^{-4}t)N$$

$0 \le t \le 100$ years

Change unit of time from 1 to 4 years so $t = 4\tau$; $d/dt = \frac{1}{4}\, d/d\tau$:

$$\frac{dN}{d\tau} = 4(8.3 \times 10^{-3} + 1.53 \times 10^{-4} \times 4\tau)N$$

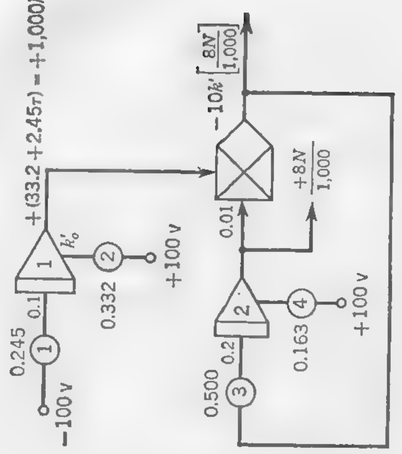
$$= (33.2 \times 10^{-3} + 2.45 \times 10^{-4}\tau)N = k'N$$

$0 \le \tau \le 25$ units of τ

Solve the analogous equation on the computer with τ as computer time in seconds so 1 sec in the computer solution represents 4 years in the problem. Let $8N/1{,}000$ be the computer voltage representing N in millions. Let 1,000 k' be the computer voltage representing k'. The computer equation becomes

$$\left(\frac{8}{1{,}000}\frac{dN}{d\tau}\right) = \frac{1{,}000\ k'}{100 \times 10}\left(\frac{8N}{1{,}000}\right)$$

$N_0 = 1{,}907$ (million)



PART 3: ELECTRONIC-ANALOG-COMPUTER TECHNIQUES FOR THE DESIGN OF SERVO SYSTEMS

By JOHN MCLEOD

ANALOG REPRESENTATION OF SERVO SYSTEMS

5.3.1. Servo System and Computer Analogy.[1] A basic *feedback control system* controls an output variable x_0 (shaft displacement, voltage, temperature) by a power source (motor, amplifier, heater) which is actuated or controlled by a suitable function of the difference between a desired input variable and the actual output (*error*). A feedback control system is usually referred to as a *servomechanism* if some mechanical actuator is regarded as the main feature.

Figure 5.3.1 shows a typical servo loop in schematic and in block-diagram form. Each block represents physical relations between system variables; these relations are usually expressed as equations and/or differential equations relating the variables. Figures 5.3.1*c* and 5.3.2 show a simple (linearized) and a more elaborate computer representation of the same servomechanism. The possibility of establishing corresponding relations, block by block, between voltages (machine variables) representing the system variables makes the electronic analog computer an attractive model for control-system analysis and synthesis.

If a sufficient number of computing elements is available, one does not usually reduce the system differential equations to their simplest form; instead, one represents specific system components (motor, valve, pickoff, etc.) by corresponding blocks of computing elements so as to preserve the block-diagram analogy. This simplifies system changes and aids the designer's intuition.

If the system is represented on a one-to-one time scale (*simulation* in the strict sense of the word), one may try actual system components by inserting them into the computer loops (*partial system test, physical simulation;* see also Sec. 5.6).

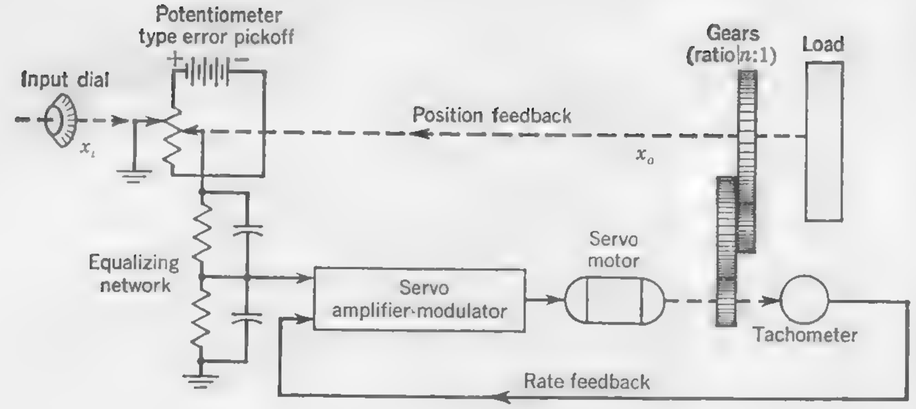
5.3.2. Linear Servomechanisms. Transfer functions of linearized servo components (equalizing networks, motors, valves, accelerometers, etc.) appear as product ratios of terms like $(\tau s + 1)$ and $(\omega_n^2 s^2 + \zeta\omega_n s + 1)$; as a further example, Fig. 5.3.3 shows the linearized representation of a typical hydraulic servomechanism. *Corresponding blocks of integrators, summing amplifiers, and coefficient potentiometers may be obtained directly from Table* 2.2.5, *or direct-analog operational amplifiers may be designed with the aid of Tables* 2.2.2 to 2.2.4 (see also Sec. 2.2).[2]

In linear-system analysis, analog computation does not replace pole-zero methods but is invaluable for subsystem frequency-response determination and for quick parameter optimization. Transient-response runs will show up dynamic properties such as excessive overshoot and can be set up so as to compute a system measure of effectiveness to be maximized by successive parameter changes (Art. 5.3.5).

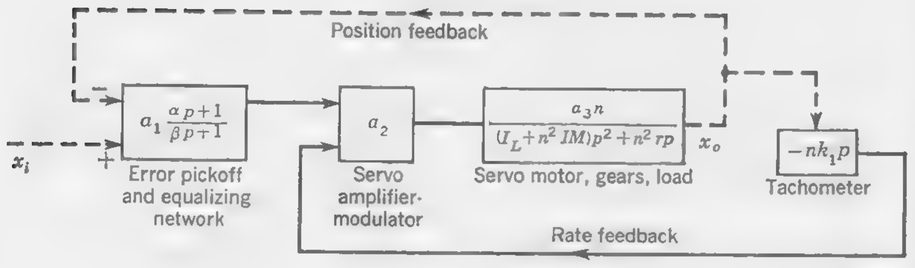
The frequency response (amplitude and phase response) of simulated servo systems and subsystems is easily obtained with the aid of a test oscillator (Sec. 3.3), or a weighting-function method (Sec. 6.6) may be used for Fourier analysis of the impulse response of a linear servo system.

5.3.3. Nonlinear Systems. Analog-computer methods are particularly important in connection with the design of *nonlinear control systems*, few of which are accessible to practical analytical design methods. Nonlinearities in control systems not only are unavoidable, but the most effective control systems are often necessarily nonlinear (Fig. 5.3.4).

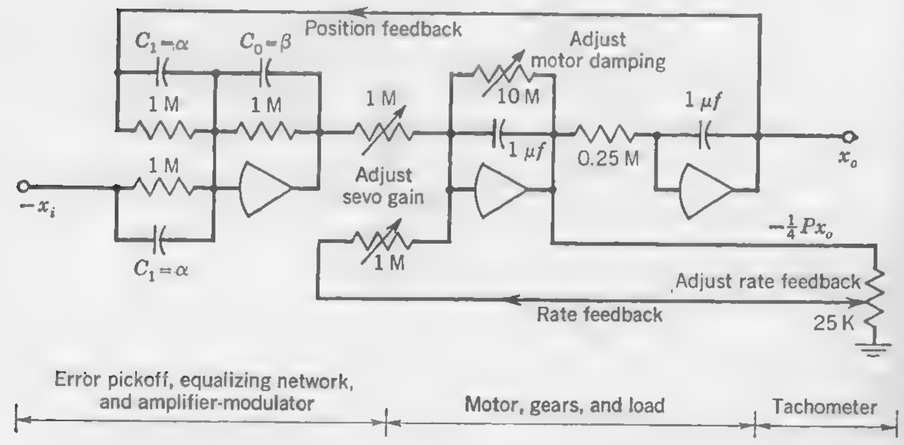
Control-system nonlinearities may be represented by nonlinear differential equations (e.g., nonlinear damping, nonlinear torque-speed-voltage characteristics of electric motors[1]) or simply by nonlinear input-output relations (transfer characteristics) of control-system components (voltage, speed, and torque limits; valve-travel/flow characteristics, gears with backlash, instruments with dead space, etc.). Section 5.4 lists analog-computer representations of typical nonlinear transfer characteristics by



(*a*)



(*b*)



(*c*)

FIG. 5.3.1. Schematic representation (*a*), linearized-system block diagram (*b*), and analog-computer setup (*c*) for a d-c servomechanism employing an amplifier-modulator, induction motor, rate feedback, and equalizing networks. Note the analogy between system components and blocks of computing elements. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

simple function generators ("fabricated" diode circuits and operational relays, see also Sec. 3.3).[1,2,3] Figures 5.3.2 to 5.3.4 show analog-computer setups for complete nonlinear servomechanisms, and Figs. 5.3.5 to 5.3.7 illustrate additional nonlinear simulation circuits.

5.3.4. Sampled-data Systems. Linear and nonlinear sampled-data control systems are represented easily and accurately by insertion of sample-hold circuits at appropriate points of the computer setup (Fig. 5.3.8).[4,5,6] Substitution of an electronic switch for the sampling relay in Fig. 5.3.8 permits higher computing speeds; computer implementation of first-order and higher-order sample-hold circuits is only slightly more

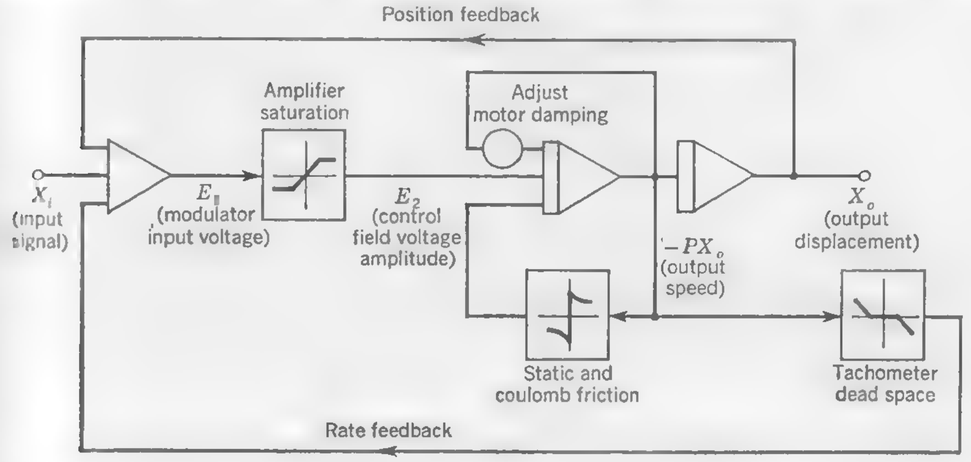


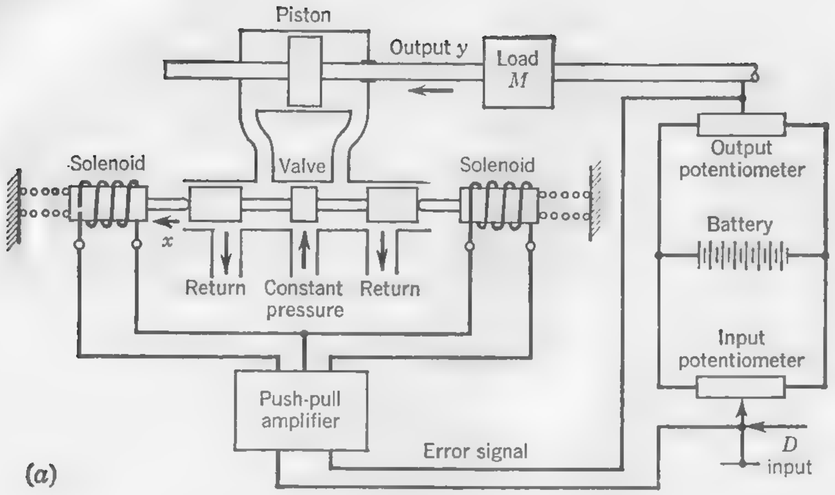
FIG. 5.3.2. More elaborate computer representation of the servomechanism of Fig. 5.3.1. Simple blocks of computing elements (Table 5.3.1) are combined to account for complicated nonlinear phenomena. (*From J. G. Truxal, Control Engineers' Handbook, McGraw-Hill Book Company, Inc., New York, 1958.*)

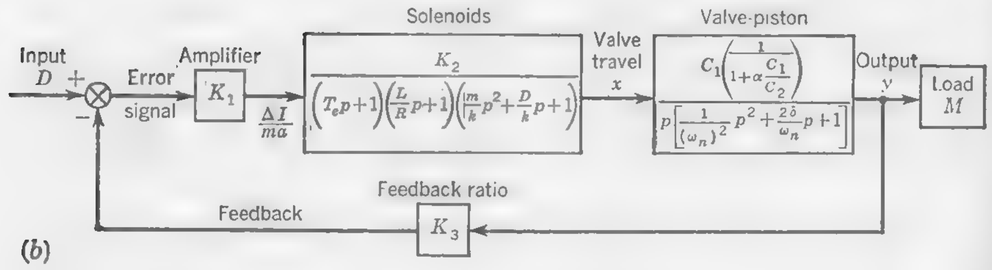
complicated but is rarely needed. Analog-computer sample-hold circuits are further discussed in Sec. 6.3.

PERFORMANCE CRITERIA

5.3.5. Evaluation of Computer Data. Analog simulation yields qualitative and quantitative insight into the effects of design-parameter changes on servo stability and on transient records of servo output and error for various inputs. In addition, runs with sinusoidal input voltages derived from external or analog-simulated low-frequency oscillators (Sec. 3.3) yield frequency-response data useful for both analysis and synthesis. In the end, the process of parameter optimization (Sec. 5.11) requires one to compare sets of servo design parameters (gain, damping, equalization-network parameters, etc.) in terms of a quantitative *measure of effectiveness* which reflects the utility of the control system in the context of a specific application. Such measures can be computed directly in the course of one or more computer runs with the simulated control system.

5.3.6. Steady-state Displacement Error, Velocity Error, and Acceleration Error.[7] Analog subtraction of voltages representing the servo output and input yields a representation of the servo error $\epsilon(t)$. The *static displacement error* $1/K_p$, *static velocity error* $1/K_v$, and *static acceleration error* $1/K_a$ are then obtained as the respective steady-state errors following sudden application of the inputs 1, t, and $t^2/2$. The corresponding reciprocal values K_p, K_v, K_a are called *static error coefficients*, while $K_1 = 1 + K_p$, $K_2 = K_v$, and $K_3 = K_a$ are the "dynamic" or *steady-state error coefficients*. Reference 7 relates servo transient response for various systems and inputs to values of K_0, K_1, and K_2.





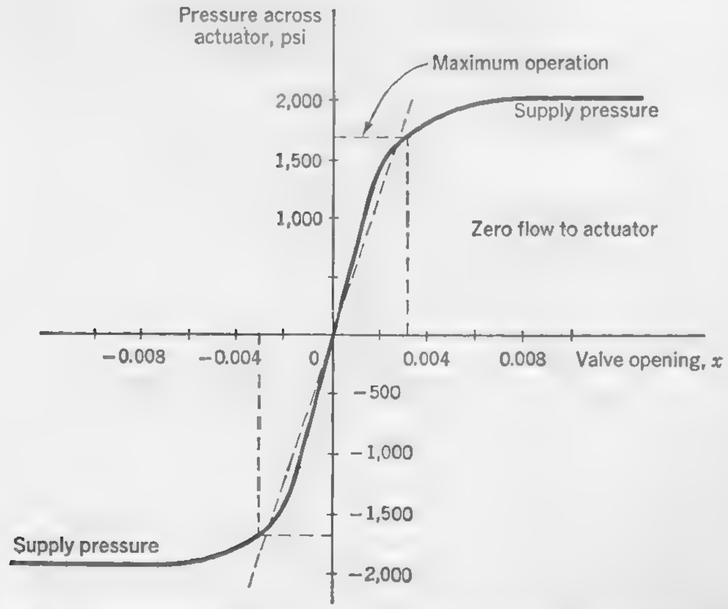


FIG. 5.3.3. Hydraulic servomechanism (*a*), and linearized-system representation (*b*). The analog-computer setup for each block is obtained directly from table 2.2.5, or direct-analog operational amplifiers can be used (Sec. 2.2). A more accurate (nonlinear) computer simulation would represent the displacement-pressure characteristic of the valve (Fig. 5.3.3*c*) by a function generator. Other nonlinearities can be introduced in the manner of Fig. 5.3.2. (*Figures from J. G. Truxal, Control Engineers' Handbook, McGraw-Hill Book Company, Inc., New York, 1958.*)

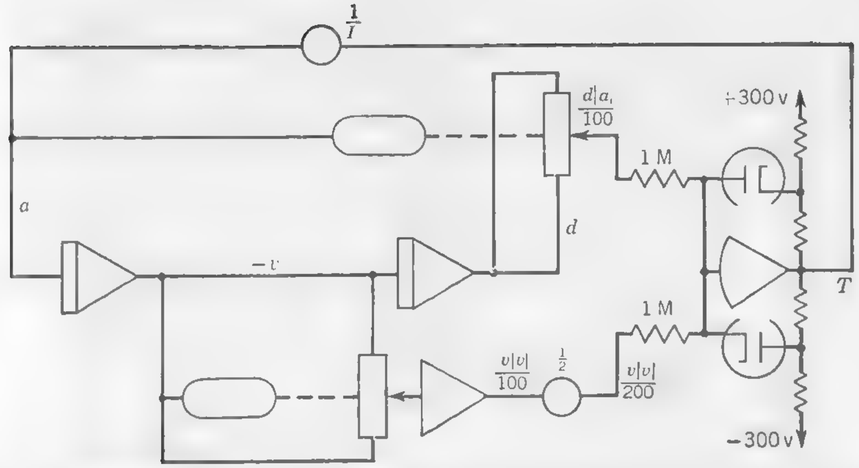


FIG. 5.3.4*a*. Simulation of a simple maximum-effort servo which applies only full positive torque T or full negative torque $-T$ for fastest possible return to zero from given initial conditions of displacement and velocity. Torque is reversed when $v|v| = -2d|a|$, where a, v, and d denote, respectively, the angular acceleration, velocity, and displacement.

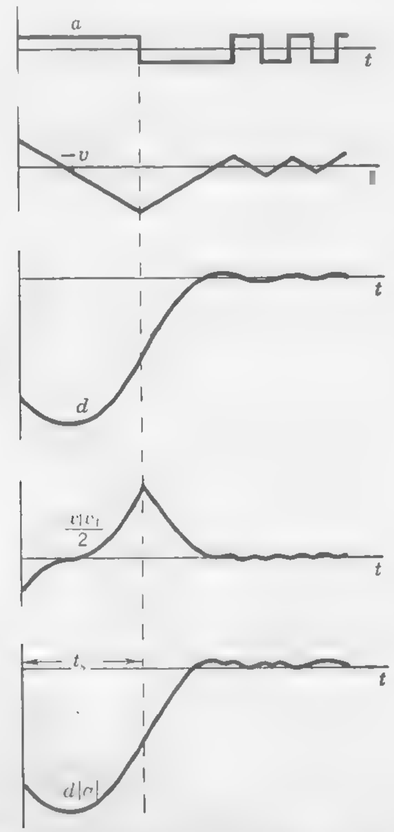


FIG. 5.3.4*b*. Records of angular acceleration a, velocity v, and displacement d for the system of Fig. 5.3.4*a*. The parameters $v|v|/2$ and $d|a|$ used to determine the switching time t_s are also recorded. In practice, one would avoid the small oscillations at the end of the run by turning the servo off in the vicinity of zero displacement.

At times a weighted sum $\lambda = a/K_p + b/K_v + c/K_a$ may be a useful measure of servo performance. In the case of *linear time-invariant* computer setups, λ can be obtained directly as the steady-state error following application of the input $a + bt + ct^2/2$, or by analogous integration of the step-function error.

5.3.7. Other Measures of Servo Performance. Application of the computer voltage representing the servo error $\epsilon(t)$ to an absolute-value circuit (Sec. 3.3) or a

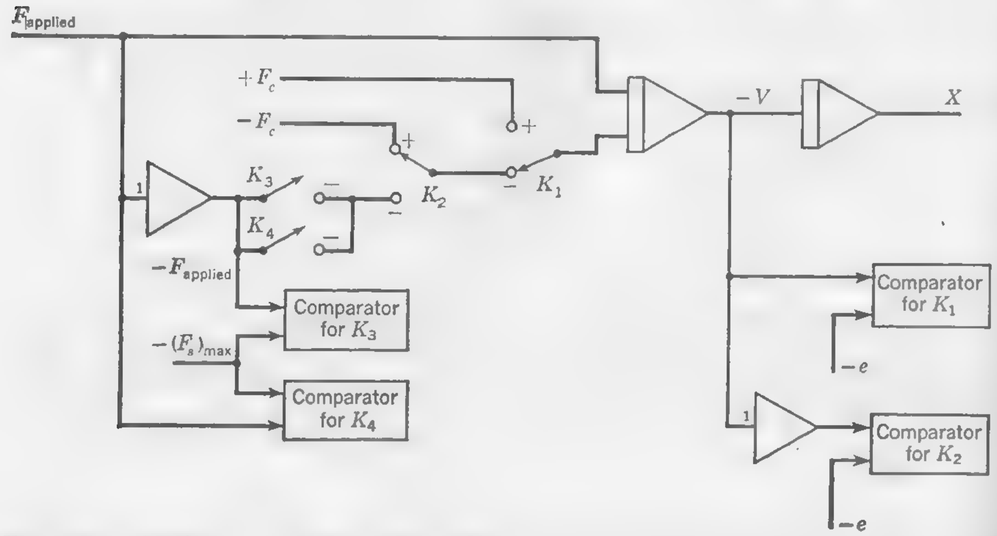


FIG. 5.3.5. Simulation of static-friction force F_s and Coulomb-friction force F_c by means of relay comparators. Each relay contact moves in the direction indicated when the algebraic sum of its comparator inputs becomes positive (see also Sec. 3.3). The static-friction force F_s cancels the applied force $F_{applied}$ if and only if $|F_{applied}| < (F_s)_{max}$ and $|v| < e$, where e is a small voltage introduced to ensure positive relay action.

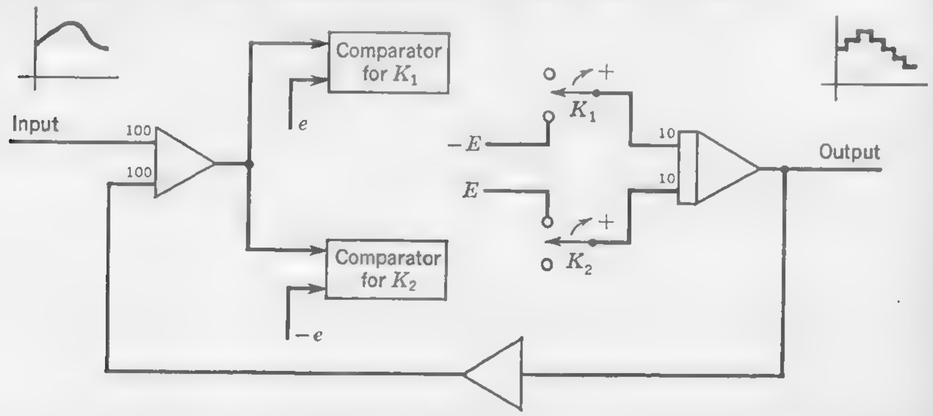
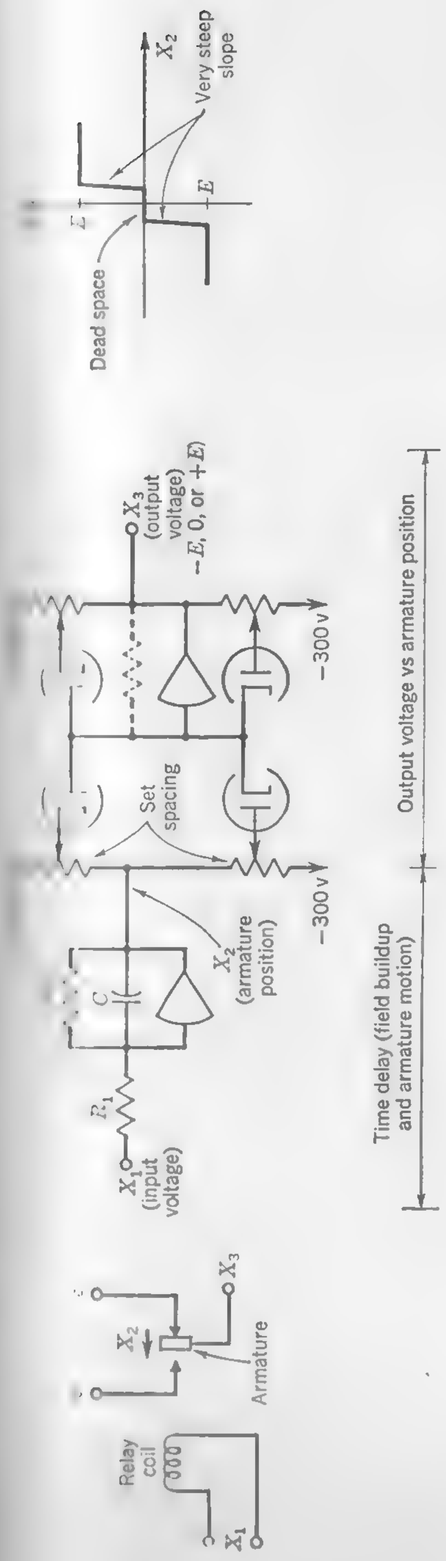


FIG. 5.3.6. Simulation of control-element granularity (e.g., potentiometer granularity, see also Table 5.4.1). Relays are actuated if the algebraic sum of the corresponding comparator inputs is positive. Granularity becomes coarser as E increases.

squaring function generator and subsequent integration yield the important performance measures

$$e_1 = \int_0^T |\epsilon(t)|\, dt \qquad e_2 = \int_0^T \epsilon^2(t)\, dt$$

which must be computed for a "typical" servo input or averaged over a set of computer runs with different inputs. The related subject of *statistical performance measures*, which also permit one to assess the performance of control systems subject to noise and/or disturbing forces, is discussed in Sec. 5.9.



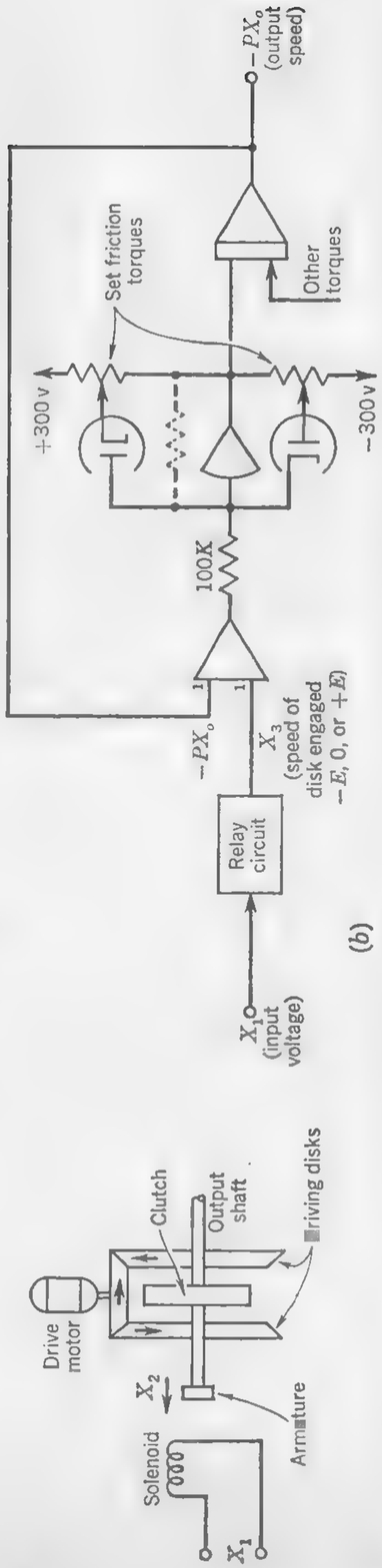


FIG. 5.3.7*a*. Representation of a polarized double-throw relay. The delays involved in field-current buildup and armature motion have been approximated by the transfer function

$$\frac{X_1}{X_2} = -\frac{R_0}{R_1}\,\frac{1}{R_0CP + 1}$$

This approximation is not unreasonable for well-designed relay circuits and sufficient armature damping. A more accurate representation involves simulation of the spring-restrained armature with damping and stops in the manner of Fig. 5.4.1.

FIG. 5.3.7*b*. Representation of a solenoid-operated dry-disk clutch. The operating delays have been approximated as in Fig. 5.3.7*a*. The dry-friction torque is developed from the difference between the output speed and the speed of the driving disk engaged.

Minimum cycling is sometimes used as a performance criterion for control systems yielding zero steady-state error. One can, thus, count the number of times, or the total length of time, that the error exceeds a specified absolute value before the system settles after a specified disturbance. This performance measure can be read from a transient record of the error, or the slicer circuit of Fig. 5.9.13 could be used for direct measurement.

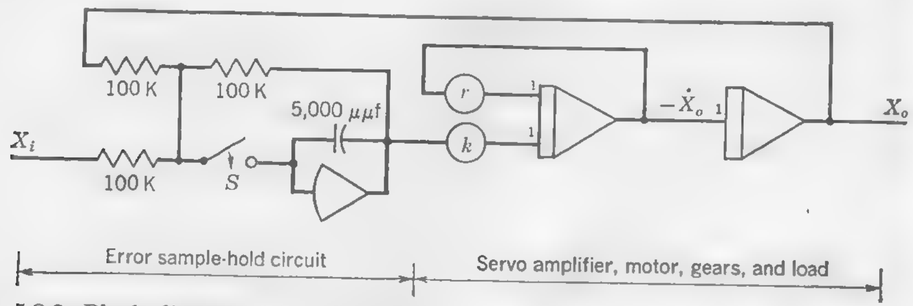


FIG. 5.3.8. Block diagram (*a*) and analog-computer setup (*b*) for a simple sampled-data servomechanism.[4] This servo setup exhibits the characteristic bandwidth limitation of a sampled-data servo as the sampling rate is decreased. The sampling switch S can be a relay driven by pulses from an analog-simulated multivibrator circuit (Table 5.4.1) or, preferably, an electronic switch (Sec. 6.3).

Finally, if the servomechanism under study forms part of a larger system, the analog computer can often be used to compute a *system performance measure* (fuel cost, miss distance, etc.).

References

1. Korn, Granino A., and T. M. Korn: *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.
2. Smith, G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1959.
3. Johnson, Clarence L.: *Analog Computer Techniques*, McGraw-Hill Book Company, Inc., New York, 1956.
4. Chestnut, H., A. Dabul, and D. Lieby: Analog-computer Study of Sampled-data Systems, *Trans. AIEE Paper* 58-1324, 1958.
5. Elgerd, O. I.: An Analog-computer Study of the Transient Behavior and Stability Characteristics of Serial-type Digital Data Systems, *Communications and Electronics*, no. 37, July, 1958.
6. Keisuke, Izawa, and L. E. Weaver: Relay-type Feedback Control Systems with Dead Time and Sampling, *Trans. AIEE Paper* 58-1026, 1958.
7. Truxal, J. G.: *Automatic Feedback Control System Synthesis*, McGraw-Hill Book Company, Inc., New York, 1955.

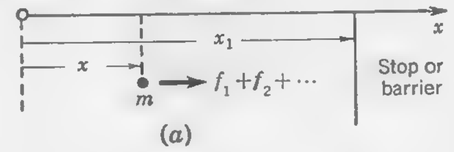
PART 4: SPECIAL CIRCUITS FOR DYNAMICAL-SYSTEM STUDIES

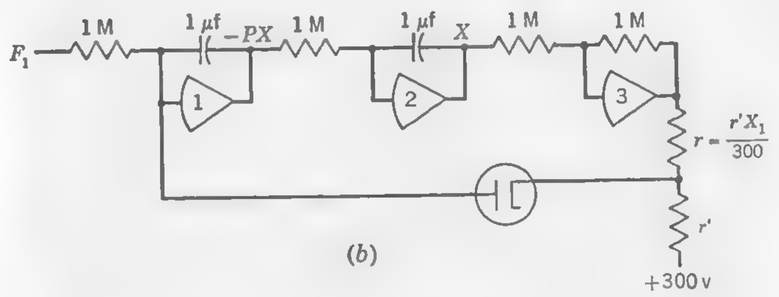
By GRANINO A. KORN *and* THERESA M. KORN

5.4.1. Representation of Nonlinear Transfer Characteristics. Diode function generators and special "fabricated" diode circuits set up on the computer patchbay can represent many types of nonlinear transfer characteristics, such as dead space in tachometers, error sensors, and measuring instruments, amplifier saturation, motor torque limits, potentiometer granularity, and valve-flow characteristics (Table 5.4.1; see also Secs. 3.3, 5.3, and 5.7).[1 to 5] Such circuits permit, in particular, the simulation of relays and clutches (Fig. 5.3.7). If high computing speed is not required, operational relays driven by comparator amplifiers (Sec. 3.3) also permit accurate simulation of nonlinear transfer characteristics; relays are particularly useful for representing

simple functions of two variables, such as static-friction forces (Fig. 5.3.5). Figure 5.3.6 shows a relay circuit for representation of granularity.

5.4.2. Limit Stops and Backlash. Direct analog-computer representation of limit stops[1,2,6] requires simulation of the forces exerted by the stop on the impinging moving





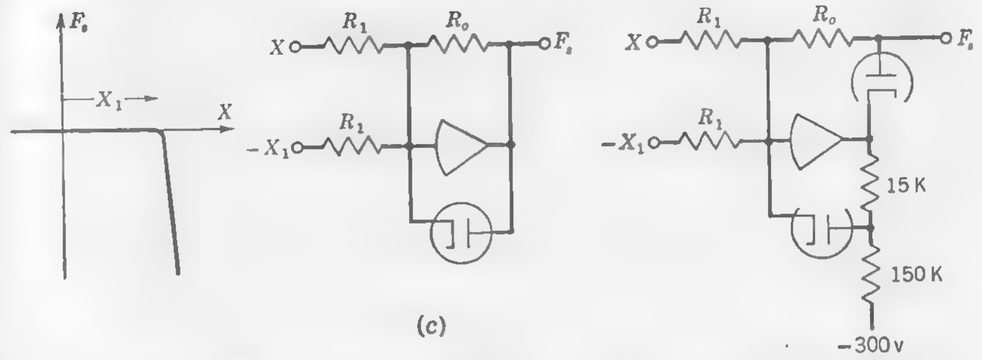


FIG. 5.4.1*a*. Linear motion of a particle restrained by a fixed or movable stop.
FIG. 5.4.1*b*. Representation of the motion for an elastic stop whose stiffness is represented by $K_S = 1/(r + r_D)$, where r_D is the diode forward resistance.
FIG. 5.4.1*c*. Representation of the elastic-stop force f_S by means of high-gain limiter circuits. The stop stiffness is represented by $K_S = R_0/R_1$.

body as the latter penetrates the stop slightly. These forces may be elastic in nature (*elastic stop*, energy is conserved), i.e., the force f_S exerted by the stop is given by

$$f_S = -k_S(x - x_1) \qquad (x > x_1) \tag{5.4.1}$$

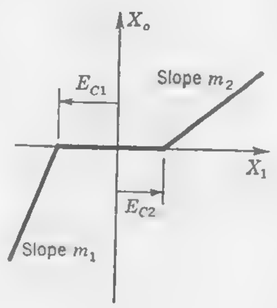
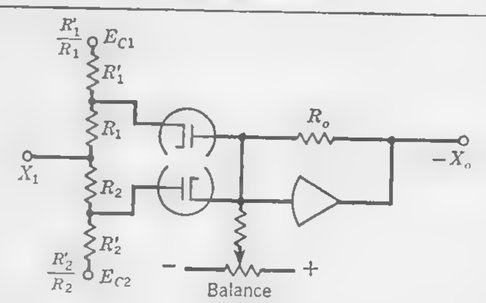
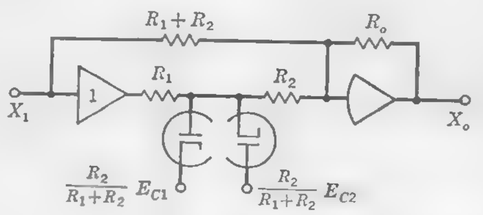
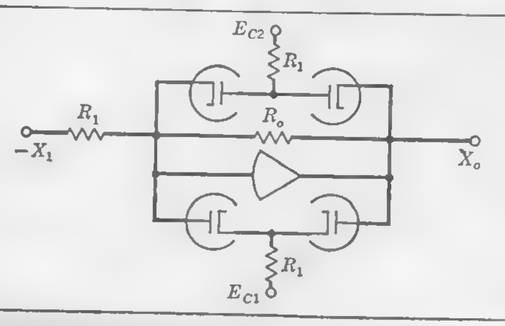
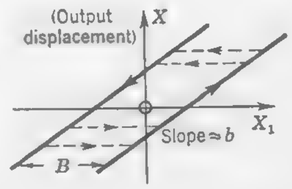
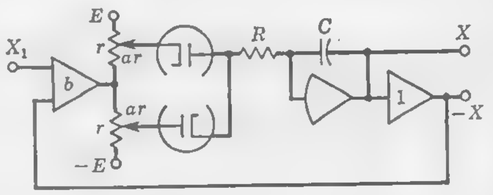
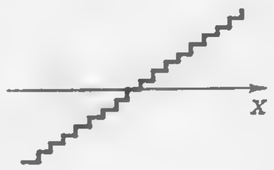
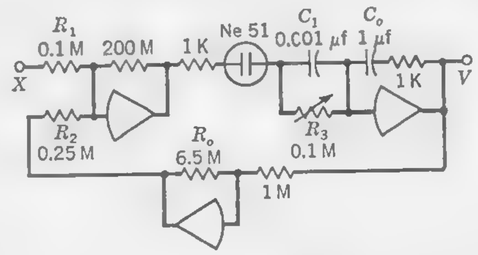
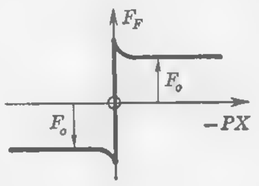
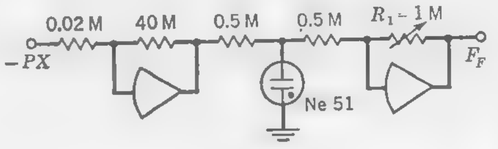
where $x - x_1$ is the penetration distance (Fig. 5.4.1*a*), or the force f_S may be approximated by a viscous force

$$f_S = -C_S \frac{d}{dt}(x - x_1) \qquad (x > x_1) \tag{5.4.2}$$

if $(d/dt)(x - x_1) > 0$ (*inelastic stop*). If the force (2) is not zero for $(d/dt)(x - x_1) < 0$, the moving body will "stick" to the stop. Combinations of the forces (1) and (2) result in *semielastic stops* with different coefficients of restitution, i.e., with partial dissipation of the initial kinetic energy.

Figures 5.4.1 and 5.4.2 show diode circuits for the representation of elastic and inelastic stops. A different type of circuit is useful for representation of an

Table 5.4.1.

Application	Transfer characteristic, or function generated	Circuit block diagram	Remarks
Representation of dead space (inert zone) in springs, control characteristics, etc. See Sec. 3.3 for additional dead-space circuits.			$m_1 = \frac{R_o}{R_1}$ $m_2 = \frac{R_o}{R_2}$
			$m_1 = m_2 = \frac{R_o}{R_1+R_2}$ Use in older computers with permanently installed shunt limiters. Requires accurate resistance matching.
			$m_1 = m_2 = \frac{R_o}{R_1}$ $(E_{C_2} = -E_{C_1})$ Easy to adjust
Approximate representation of backlash in gears, linkages, process-control error detectors, etc. (assuming inelastic stops).			$B \approx \frac{2aE}{b(1-a)}$ $\quad (ar+R)C \ll 1$ Typical values: $r = R = 50\,K$ $C = 1\ \mu f$ $b = 1$ $E = 200$ v
Approximate representation of granularity (e.g., potentiometer granularity).†‡ See also Sec. 5.3			Average slope $= \frac{R_2}{R_o R_1}$ Step height $\approx \frac{R_2}{20}$ R C controls the step rise time and thus the permissible stepping rate. For a more accurate presentation, a pair of thyratrons or diodes may replace the neon bulb.†
Very rough representation of starting friction and coulomb friction. See also Sec. 5.3			$F_o \approx 130 R_o$ volts

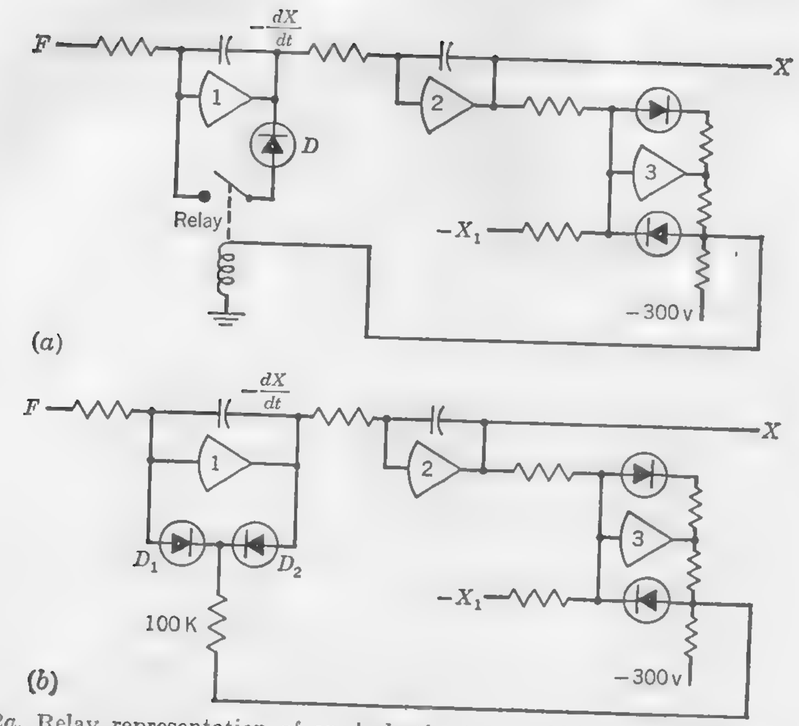


FIG. 5.4.2*a*. Relay representation of an *inelastic* stop at $X = X_1$. The diode prevents "sticking."

FIG. 5.4.2*b*. Diode representation of an inelastic stop (from Ref. 6). Note that the diode switch passes current in one direction only and thus prevents sticking.

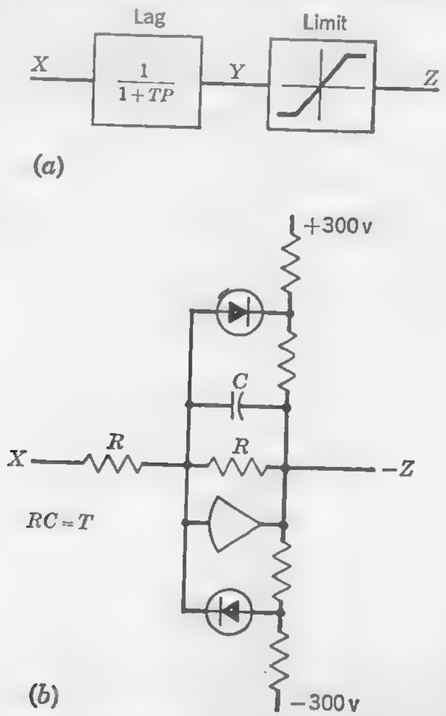


FIG. 5.4.3. Block diagram showing an amplitude-limited quantity Z preceded by a simple lag.

amplitude-limited quantity Z preceded by a simple lag (Fig. 5.4.3). If the moving body is limited to an interval between two stops ($x_1 \geq x \geq x_2$), the circuits of Figs. 5.4.1 and 5.4.2 may be duplicated with appropriately reversed diodes and bias voltages, or a dead-space circuit can be used. The circuits of Figs. 5.4.1 and 5.4.2 permit many other variations and generalizations. In particular, the stops themselves may be mov-

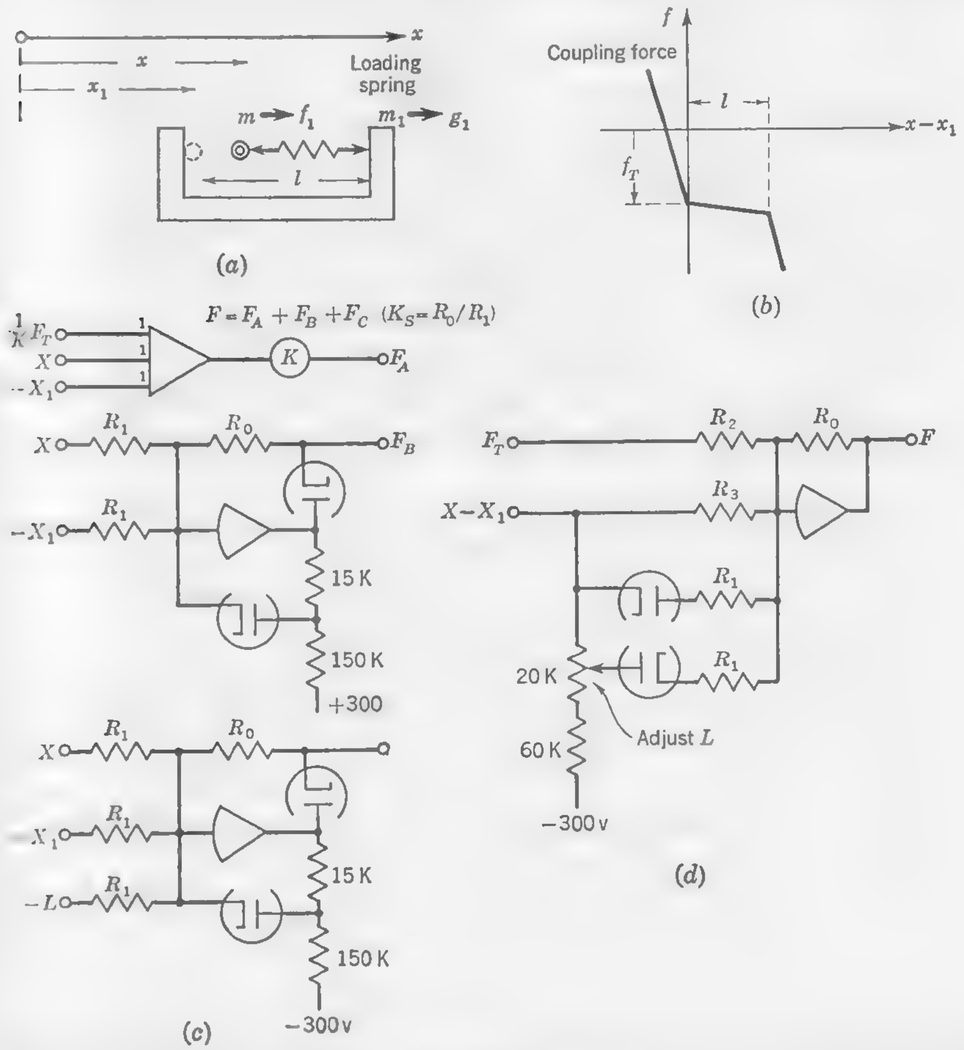


FIG. 5.4.4. Representation of the coupling force associated with spring-loaded linkages, cams, or gears subject to ordinary backlash (the special case $K = F_T = 0$ corresponds to ordinary backlash, without a loading spring). This example illustrates action and reaction of perfectly elastic stops. Note that the backlash circuit in Table 5.4.1 assumes inelastic stops and does not represent the force on the stop; it is thus more suitable for simulation of instrument backlash than gear backlash. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

ing bodies, so that x_1 becomes a variable. Since the impact force on the stop will be equal and opposite to the force represented by f_S in Fig. 5.4.1, the computer can easily deal with motions involving impacts of two or more rigid bodies. In this manner, one can simulate *elastic backlash* (Fig. 5.4.4; see Table 5.4.1 for an approximate representation of *inelastic backlash*). In the case of *oblique impacts* on smooth barriers, one must consider components of the impact force in the coordinate directions. It is, for instance, possible to simulate the rebound of a billiard ball from a barrier.

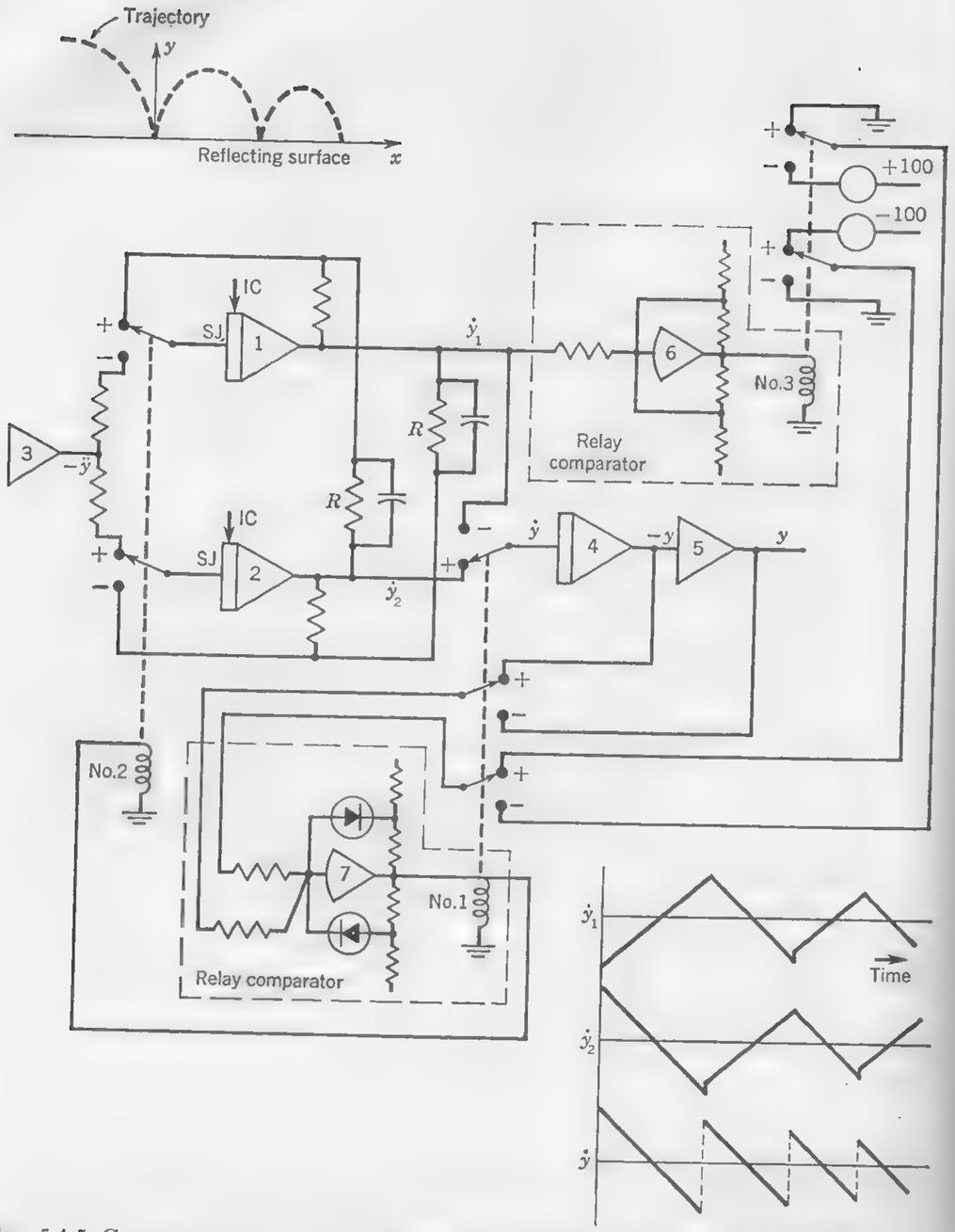


FIG. 5.4.5. Computer study of a point mass that strikes a reflecting surface: trajectory (*a*), computer schematic (*b*), and typical records (*c*). The relay comparator circuit switches each relay arm to its "plus" contact when the output of the amplifier is positive and to its "minus" contact when the output is negative. With the relays as shown, integrator 2 is computing $\dot{y}$, and integrator 1 is charged to its new initial condition. When $y = 0$, relays 1 and 2 switch and integrator 1 computes $\dot{y}$. Relay 3 is used to latch relays 1 and 2 at $y = 0$. Polarized relays are used. (*Contributed by B. D. Loveman*).

Simulation of elastic stops requires a measure of caution simply because the true impact forces may be so very large compared with other applied forces that a difficult scaling problem results. Thus in the computer setup of Fig. 5.4.1*b*, the diode is not, as in most analog-computer setups, in series with a large resistance, and diode nonlinearities and capacitance can result in substantial errors. For accurate computation, one can often slow down the time scale by representing all forces by smaller voltages, so that the circuit of Fig. 5.4.1*c* can be used with reasonable resistance values. This is, unfortunately, not possible if the computer setup is to be used on a 1:1 time scale, as for partial system tests.

Figure 5.4.5 shows a less direct and potentially more accurate method for representing elastic and semielastic stops. This computer setup employs analog storage circuits (Sec. 4.2) to represent the motion of a point mass that strikes a reflecting, frictionless surface (e.g., a bouncing ball). At the instant it strikes the surface, the normal component of the velocity $\dot{y}$ reverses sign, and if there is energy dissipation, it is also attenuated. Two similar integrator circuits are used to compute $\dot{y}$ for alternate sections of the trajectory, i.e., between the points at which the mass strikes the surface, $y = 0$. As one circuit is computing $\dot{y}$, the other integrator is charged to a new initial condition. Relay comparator circuits are used to switch the integrators. A similar circuit is used for ray-tracing studies in which the ray strikes a reflecting surface.[7]

References

1. Johnson, Clarence L.: *Analog Computer Techniques*, McGraw-Hill Book Company, Inc., New York, 1956.
2. Korn, Granino A., and T. M. Korn: *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.
3. Smith, G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1959.
4. Morrill, C. D., and R. V. Baum: Diode Limiters Simulate Mechanical Phenomena, *Electronics*, **25**: 122 (November, 1952).
5. Koerner, H., and G. A. Korn: Function Generation with Operational Amplifiers, *Electronics*, Nov. 6, 1959.
6. Korn, G. A.: Simulation of Inelastic Stops, *IRE Trans. PGEC*, March, 1960.
7. Wong, Ming S.: Refraction Anomalies in Airborne Propagation, *Proc. IRE*, September, 1958.

PART 5: AIRCRAFT, AUTOPILOT, AND MISSILE PROBLEMS

By LOUIS BAUER

RIGID-BODY MOTION OF AIRCRAFT

5.5.1. Flight Equations and Simplifying Assumptions. The motion of an aircraft as a rigid body has six degrees of freedom: translation in three directions, characterized by the *drag, lift, and side-force equations,* and rotation in three directions, described by the *roll, pitch, and yaw equations.* To these *flight equations* one must add equations describing a *control system* (inputs corresponding to pilot commands, or beam-riding systems, homing radar, etc., in pilotless aircraft and missiles). Figure 5.5.1 shows the block-diagram representation of a typical guided-missile problem.

In early studies of a new aircraft system, greatly simplified phases of the problem are tentatively isolated ("decoupled") and studied separately. Thus one may attempt to disregard rotation and study the *trajectory of the aircraft center of gravity* separately, or one may investigate *rotational stability* without regard to translatory motion. Again, one may begin with a *longitudinal study* restricted to motion in a vertical plane (lift and pitch equations), and a *lateral study* of motion in a horizontal plane (side-force and yaw equations). Roll is then added to the latter motion, and the effects of velocity changes due to thrust and drag (drag equation) may be added to both longitudinal and lateral studies. Note, however, that late-model aircraft may

exhibit such strong interaction between degrees of freedom that decoupled flight equations are useful only as a very rough guide.

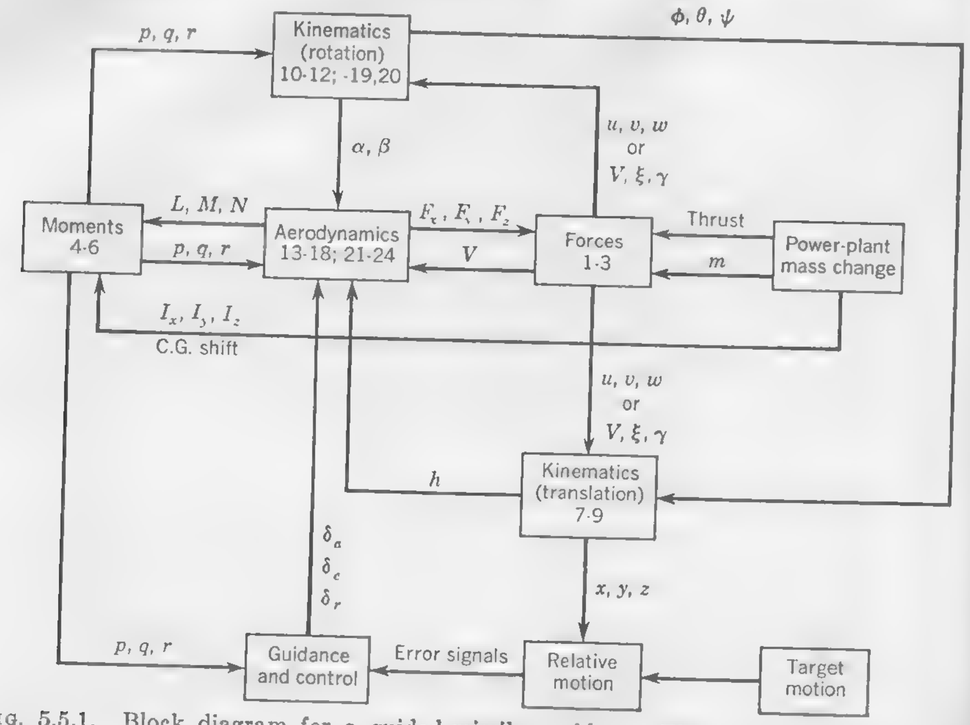


FIG. 5.5.1. Block diagram for a guided-missile problem. The numbers shown in the blocks refer to the numbers of the equations in the text. No equations for the guidance and control system have been listed because of the great variety of possibilities. In the case of piloted aircraft the inputs will be the aileron, elevator, and rudder deflections, as well as the throttle setting controlling the thrust. The equations giving the variations of mass and moments of inertia with fuel consumption and the expressions for the moments due to center-of-gravity shift will also have to be added when such effects are appreciable.

5.5.2. Coordinate Systems. The form of the equations of motion depends upon the choice of the coordinate system. The following systems are widely used:

Earth Coordinate System (ECS)

1 axis (x axis): north
2 axis (y axis): east
3 axis (z axis): down

Body Coordinate System (BCS, Body Axes)

1 axis (x axis): aircraft longitudinal axis
2 axis (y axis): perpendicular to the x axis, toward the right wing
3 axis (z axis) perpendicular to the x and y axes, forming a right-handed system, i.e., in the vertical symmetry plane and down

Velocity Coordinate System (VCS, Wind Axes)

1 axis (x axis): in the direction of the velocity vector
2 axis (y axis): perpendicular to the x and z axes, forming a right-handed system
3 axis (z axis): perpendicular to the x axis, in the airplane vertical symmetry plane, positive "down"

The *VCS* described here is tied in with the velocity vector and the airplane symmetry plane. Another velocity coordinate system is also used:

Velocity-Earth Coordinate System (VECS)

1 axis (x axis): in the direction of the velocity vector
2 axis (y axis): perpendicular to the velocity vector, in a horizontal plane
3 axis (z axis): perpendicular to the x and y axes, forming a right-handed system

The two velocity coordinate systems differ only by a rotation about the velocity vector; therefore the first two angles defining these two systems are identical, namely, the heading angle and the angle of climb.

The ECS is useful in computations of aircraft position (*kinematic equations of translation*); the BCS is useful for expressing the moment equations (*dynamic equations of rotation*).

The VCS and VECS are useful in computations of flight-path angles; the aircraft angles of attack appear as angles between BCS and VCS axes. Since the expressions for the angles between the BCS and VECS axes in three dimensions are very complex, the VCS has been largely abandoned for three-dimensional problems, and the angles of attack are computed from the velocity components expressed in the BCS. Since wind-tunnel data (aerodynamic coefficients) are naturally obtained in VCS, they now have to be converted into BCS.

Other coordinate systems tied to the aircraft's radar, gyros, or other control-system components are also frequently used and will depend on specific control-system configurations. Note also that high-speed aircraft and long-range missiles may require one to modify flight equations expressed in the ECS to allow for the earth's rotation, since the ECS is not a true inertial system.

5.5.3. Typical Three-dimensional Flight Equations (see Table 5.5.1 for symbols).

Dynamic Equations for Translation

$$m\dot{u} = F_X + T - mg\sin\theta - mqw + mrv \tag{5.5.1}$$
$$m\dot{v} = F_Y + mg\cos\theta\sin\phi - mru + mpw \tag{5.5.2}$$
$$m\dot{w} = F_Z + mg\cos\theta\cos\phi - mpv + mqu \tag{5.5.3}$$

Table 5.5.1. Glossary of Symbols

Symbol	*Definition*
A_w	Wing area
C_L, C_M, C_N	Roll, pitch, and yaw moment coefficients. The added subscripts refer to the source of these moments
C_X, C_Y, C_Z	Drag, side force, and lift coefficients. The added subscripts refer to the source of these forces
F_X, F_Y, F_Z	Aerodynamic force components in body-axis system
F_X', F_Y', F_Z'	Aerodynamic force components in wind-axis system
I_X, I_Y, I_Z	Moments of inertia with respect to principal axes
L^d, M^d, N^d	Components of the total external dynamic moment
L^s, M^s, N^s	Components of the total external static moment
M	Mach number
T	Thrust
V	Velocity
V_s	Velocity of sound
b	Semiwing span
c	Mean aerodynamic chord
g	Acceleration due to gravity
h	Altitude
m	Mass of aircraft
p, q, r	Angular-velocity components in body-axis system
u, v, w	Velocity components in body-axis system
x, y, z	Coordinates in the ECS
α	Angle of attack in pitch
β	Angle of attack in yaw (angle of side slip)
δ_a, δ_e, δ_r	Deflections of aileron, elevator, and rudder, respectively
ϕ, θ, ψ	Roll, pitch, and yaw angles
ρ	Density of air
ξ	Heading angle (flight-path angle in azimuth)
γ	Angle of climb (flight-path angle in elevation)

Dynamic Equations for Rotation

$$I_X\dot{p} + (I_Z - I_Y)qr = L^s + L^d \tag{5.5.4}$$
$$I_Y\dot{q} + (I_X - I_Z)rp = M^s + M^d \tag{5.5.5}$$
$$I_Z\dot{r} + (I_Y - I_X)pq = N^s + N^d \tag{5.5.6}$$

Kinematic Equations for Translation (Coordinate Transformation from BCS to ECS)

$$\dot{x} = u\cos\theta\cos\psi - (v\cos\phi - w\sin\phi)\sin\psi + (v\sin\phi + w\cos\phi)\sin\theta\cos\psi \tag{5.5.7}$$

$$\dot{y} = u\cos\theta\sin\psi + (v\cos\phi - w\sin\phi)\cos\psi + (v\sin\phi + w\cos\phi)\sin\theta\sin\psi \tag{5.5.8}$$

$$\dot{z} = -u\sin\theta + v\cos\theta\sin\phi + w\cos\theta\cos\phi \tag{5.5.9}$$
$$\dot{h} = -\dot{z} \tag{5.5.9a}$$

Kinematic Equations for Rotation

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta \tag{5.5.10}$$
$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{5.5.11}$$
$$\dot{\psi} = (q\sin\phi + r\cos\phi)\sec\theta \tag{5.5.12}$$

Aerodynamic Forces

$$F_X = \tfrac{1}{2}\rho V^2 A_w C_X(\alpha, M) \tag{5.5.13}$$
$$F_Y = \tfrac{1}{2}\rho V^2 A_w (C_{Y_\beta}\beta + C_{Y_{\delta_r}}\delta_r) \tag{5.5.14}$$
$$F_Z = \tfrac{1}{2}\rho V^2 A_w C_{Z\alpha}\alpha \tag{5.5.15}$$

Aerodynamic Moments

$$L^s = \tfrac{1}{2}\rho V^2 A_w b(C_{L_{\delta_a}}\delta_a + C_{L_{\beta_a}}\beta) \qquad L^d = \rho V A_w \frac{b^2}{4} C_{L_p} p \tag{5.5.16}$$

$$M^s = \tfrac{1}{2}\rho V^2 A_w c(C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e) \qquad M^d = \rho V A_w \frac{c^2}{4} C_{M_q} q \tag{5.5.17}$$

$$N^s = \tfrac{1}{2}\rho V^2 A_w b(C_{N_\beta}\beta + C_{N_{\delta_r}}\delta_r) \qquad N^d = \rho V A_w \frac{b^2}{4} C_{N_r} r \tag{5.5.18}$$

Miscellaneous Definitions

$$\alpha = \arctan\frac{w}{u} \tag{5.5.19}$$

$$\beta = \arctan\frac{v}{u} \tag{5.5.20}$$

$$M = \frac{V}{V_s} \tag{5.5.21}$$

$$V^2 = u^2 + v^2 + w^2 \quad \text{or} \quad V = u\sqrt{1 + \alpha^2 + \beta^2} \quad \text{or} \quad V = u \quad \text{approx} \tag{5.5.22}$$

$$V_s = V_s(h) \tag{5.5.23}$$
$$\rho = \rho(h) \tag{5.5.24}$$

β is sometimes defined according to another convention to be $-\arctan v/u$ (see discussion below).

Equations (19) and (20) are frequently replaced by the approximations

$$\alpha = \frac{w}{u} \tag{5.5.19a}$$

$$\beta = \frac{v}{u}\left(\text{or } -\frac{v}{u}, \text{ as the case may be}\right) \tag{5.5.20a}$$

5.5.4. Discussion of Equations and Alternate Formulation. The first three equations are the dynamic equations for translation, written in the BCS. The advantage of writing them in this system is that u, v, and w which are computed from them can be used directly in the computation of α and β. The disadvantage is that $\dot{u}$, $\dot{v}$, and $\dot{w}$ are *not* the accelerations (g's) felt by the pilot and plane. The last two terms on the right-hand side of each equation ($-mqw + mrv$, etc.) are the so-called *gyroscopic terms* which have to be included because the coordinate system is rotating

If the first three equations were written in the VECS they would appear as follows:

$$m\dot{V} = F_X' + T\cos\alpha\cos\beta - mg\sin\gamma \qquad \text{(DRAG)} \tag{5.5.1a}$$
$$mV\cos\gamma\cdot\dot{\xi} = F_Y' + T\cos\alpha\sin\beta \qquad \text{(SIDE FORCE)} \tag{5.5.2a}$$
$$-mV\dot{\gamma} = F_Z' - T\sin\alpha\cos\beta + mg\cos\gamma \qquad \text{(LIFT)} \tag{5.5.3a}$$

In this case Eqs. (7) to (9) become

$$\dot{x} = V\cos\gamma\cos\xi \tag{5.5.7a}$$
$$\dot{y} = V\cos\gamma\sin\xi \tag{5.5.8a}$$
$$\dot{z} = -\dot{h} = -V\sin\gamma \tag{5.5.9a}$$

The angles ξ and γ are the *flight-path angles* in yaw and pitch, i.e., the azimuth and elevation angle of the velocity vector.

The expressions appearing in Eqs. (1a) to (3a) now correspond to the actual g forces experienced by the pilot and aircraft.

For small angles α and β these equations become

$$m\dot{V} = F_X' + T - mg\sin\gamma \tag{5.5.1b}$$
$$mV\cos\gamma\cdot\dot{\xi} = F_Y' + T\beta \tag{5.5.2b}$$
$$-mV\dot{\gamma} = F_Z' - T\alpha + mg\cos\gamma \tag{5.5.3b}$$

The aerodynamic forces appearing in Eqs. (1b) to (3b) are given in Eqs. (13) to (15).

Equations (4), (5), and (6) are the dynamic equations for rotation (*Euler's equations*). It has been assumed that the body axes coincide with the principal axes of inertia. Otherwise, product-of-inertia terms also have to be included. The aerodynamic moments have been broken up into the *static moments* (*restoring moments*) and *dynamic moments* (*damping moments*), denoted by superscripts s and d. These moments are given in Eqs. (16) to (18).

The coordinates defining *aircraft position* are determined from Eqs. (7), (8), and (9). These equations express the coordinate transformations from the BCS to the ECS. Equations (7a) to (9a) perform the corresponding transformations from the VECS to the ECS.

Equations (10), (11), and (12) express the rates of change of the orientation angles ϕ, θ, and ψ in terms of the angular-velocity components p, q, and r. Note that in general it is *not* true that $\dot{\phi} = p$, $\dot{\theta} = q$, or $\dot{\psi} = r$.

The aerodynamic coefficients included in Eqs. (13) to (18) are intended to be only typical and not exhaustive.

Equations (19) and (20) give the expressions for the *angles of attack* in pitch and yaw. Since these are extremely important quantities in all aerodynamic problems, a brief description using simplified (two-dimensional) diagrams will be given here.

In Fig. 5.5.2a, $1E$ is the horizontal reference axis, $1V$ the direction of the velocity vector V, and $1B$ and $3B$ the x and z axes in the body coordinate system.

The *angle of attack* is the angle between the direction in which the airplane is going and the direction the airplane is pointing. (This is not intended to be a rigorous definition based upon fundamental aerodynamic principles, but it is useful for the applications here.)

From Fig. 5.5.2a we have

$$\alpha = \theta - \gamma \qquad \text{(TWO-DIMENSIONAL CASE ONLY)}$$

or, using the components of the velocity expressed in the BCS,

$$\alpha = \arctan\frac{w}{u}$$

The first of these two expressions applies only to the two-dimensional case. The sign convention is such that, as shown in the figure, the angle α is positive. The aerodynamic behavior of conventional airplanes is such that a positive angle α produces a *lift force*, i.e., a force at right angles to V and in the negative 3 direction, and a *restoring moment*, i.e., a moment tending to decrease α.

In the yaw plane the situation is very similar, but the sense of a positive value for β is not generally agreed upon.

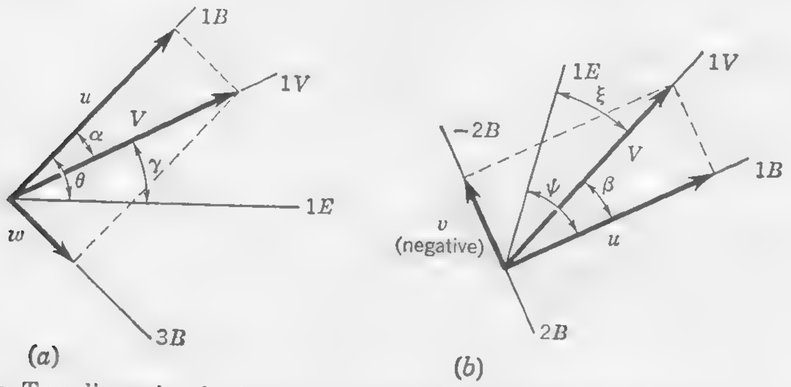


FIG. 5.5.2*a*. Two-dimensional schematic showing the angle of attack in pitch. The quantities shown here all lie in a vertical plane. The aircraft is pointing in the direction $1B$, and moves in the direction $1V$.
FIG. 5.5.2*b*. Two-dimensional schematic showing the angle of attack in yaw (sideslip). The quantities shown here lie in a horizontal plane. The aircraft is pointing in the direction $1B$ and moves in the direction $1V$.

In Fig. 5.5.2*b*, $1E$ is the earth reference (north), $1V$ the direction of the velocity vector V, and $1B$ and $2B$ the x and y axes in the BCS. If the sign convention for β is analogous to the one used for α as regards the wind axes and body axes, then

$$\beta = \psi - \xi \qquad \text{(TWO-DIMENSIONAL CASE ONLY)}$$

Since the velocity component of V in the $2B$ direction is now negative for a positive β

$$\beta = \arctan \frac{-v}{u}$$

For values of β as shown in Fig. 5.5.2*b*, the resulting aerodynamic force will be a side force, at right angles to V and in the positive 2 direction, and the moment will be a restoring moment tending to reduce β.

If, however, the angle β is defined as

$$\beta = \arctan \frac{v}{u}$$

a positive value of β will be opposite to the one shown in Fig. 5.5.2*b*, and the aerodynamic coefficients will have to have the opposite signs in order to correspond to the same physical effects.

5.5.5. Computer Flow Diagram. Figure 5.5.3 shows the computer flow diagram corresponding to Eqs. (1) to (24). No attempt has been made to scale the equations, but a sufficient number of amplifiers have been provided to make sure that all quantities are available with their proper sign. The inputs to the problem are the rudder, elevator, and aileron deflections and the thrust T.

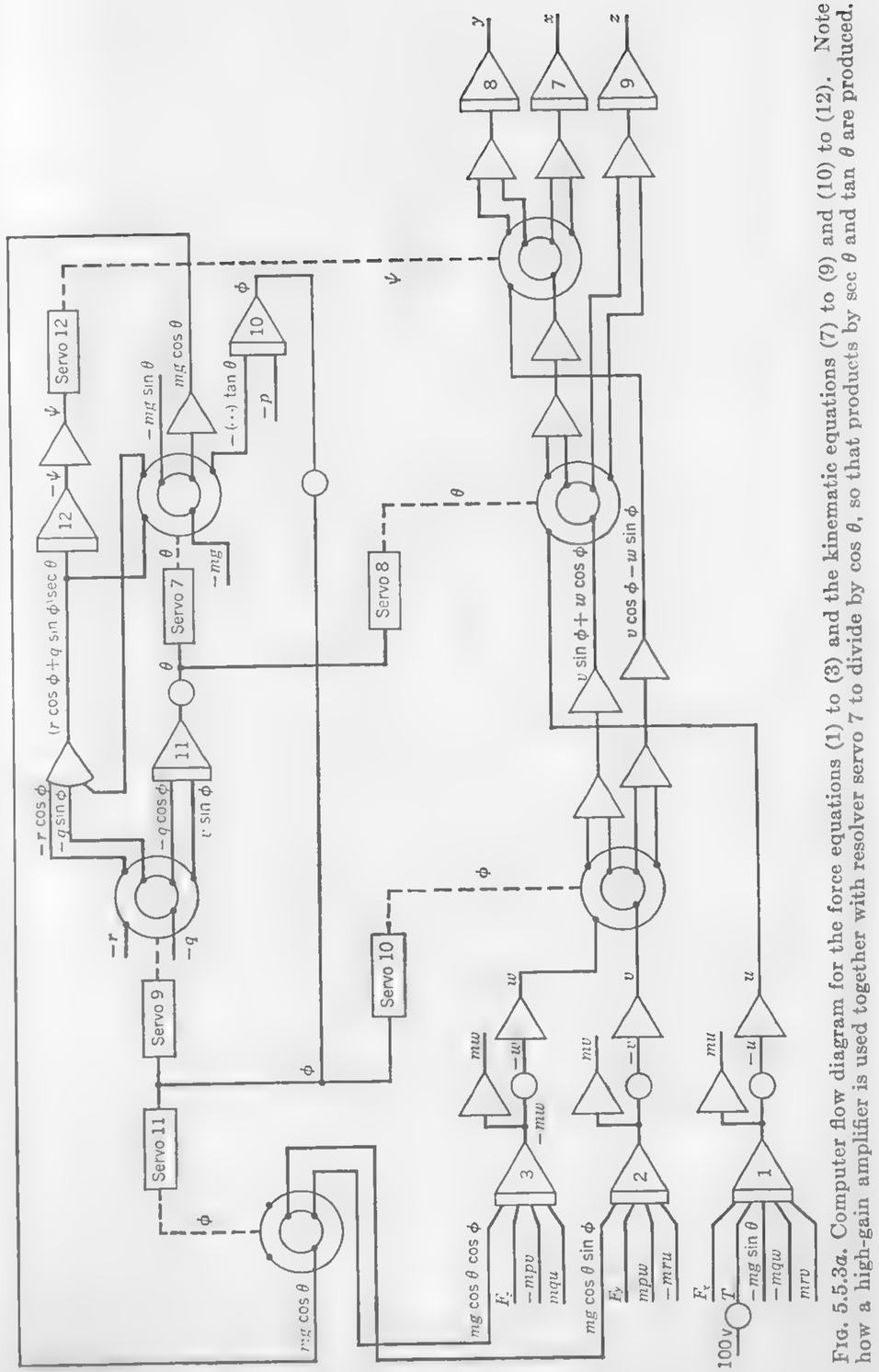


FIG. 5.5.3*a*. Computer flow diagram for the force equations (1) to (3) and the kinematic equations (7) to (9) and (10) to (12). Note how a high-gain amplifier is used together with resolver servo 7 to divide by $\cos\theta$, so that products by $\sec\theta$ and $\tan\theta$ are produced.

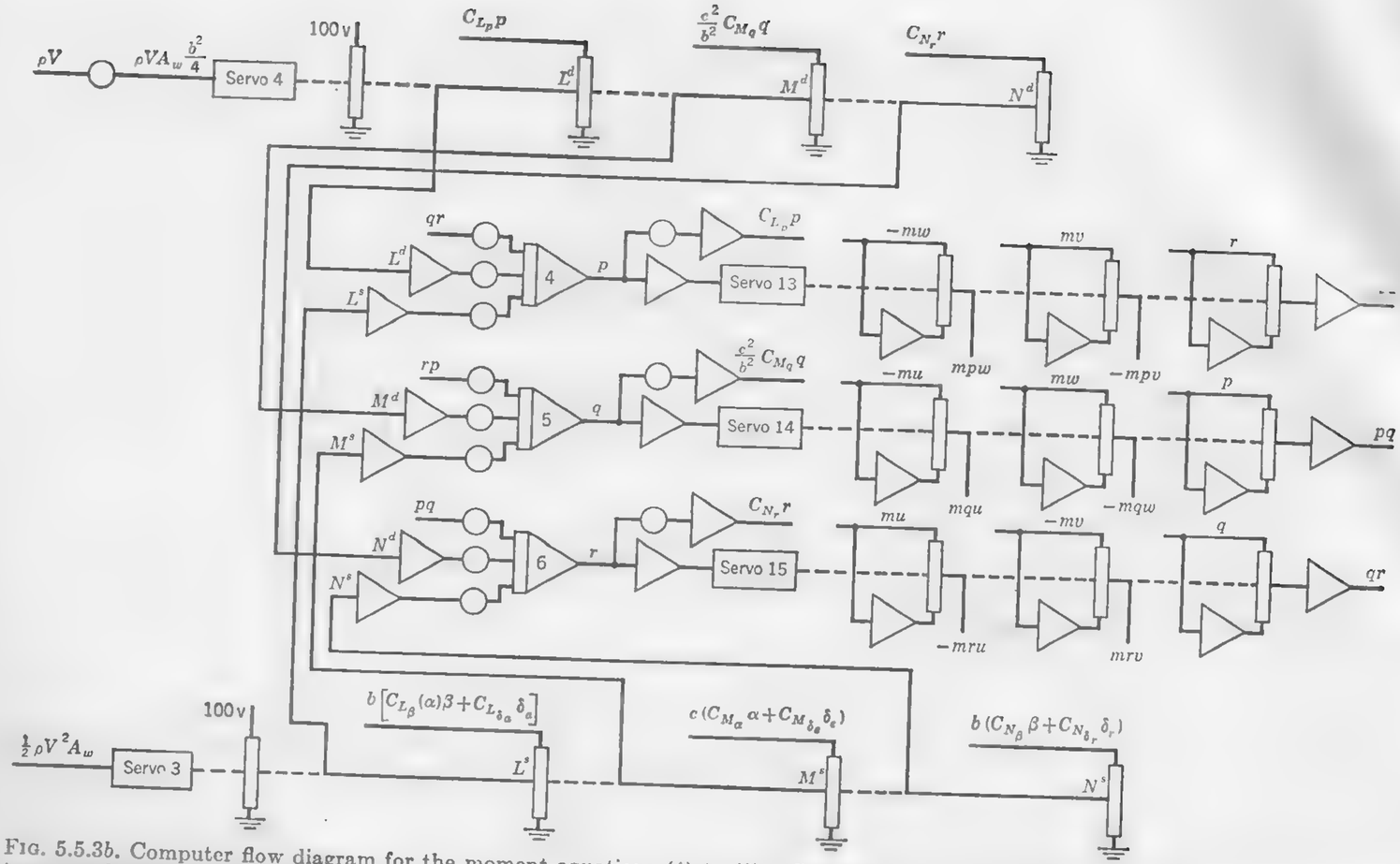


FIG. 5.5.3b. Computer flow diagram for the moment equations (4) to (6). Servos 13, 14, and 15 yield gyroscopic and cross-coupling terms, and servos 3 and 4 compute static and dynamic moments.

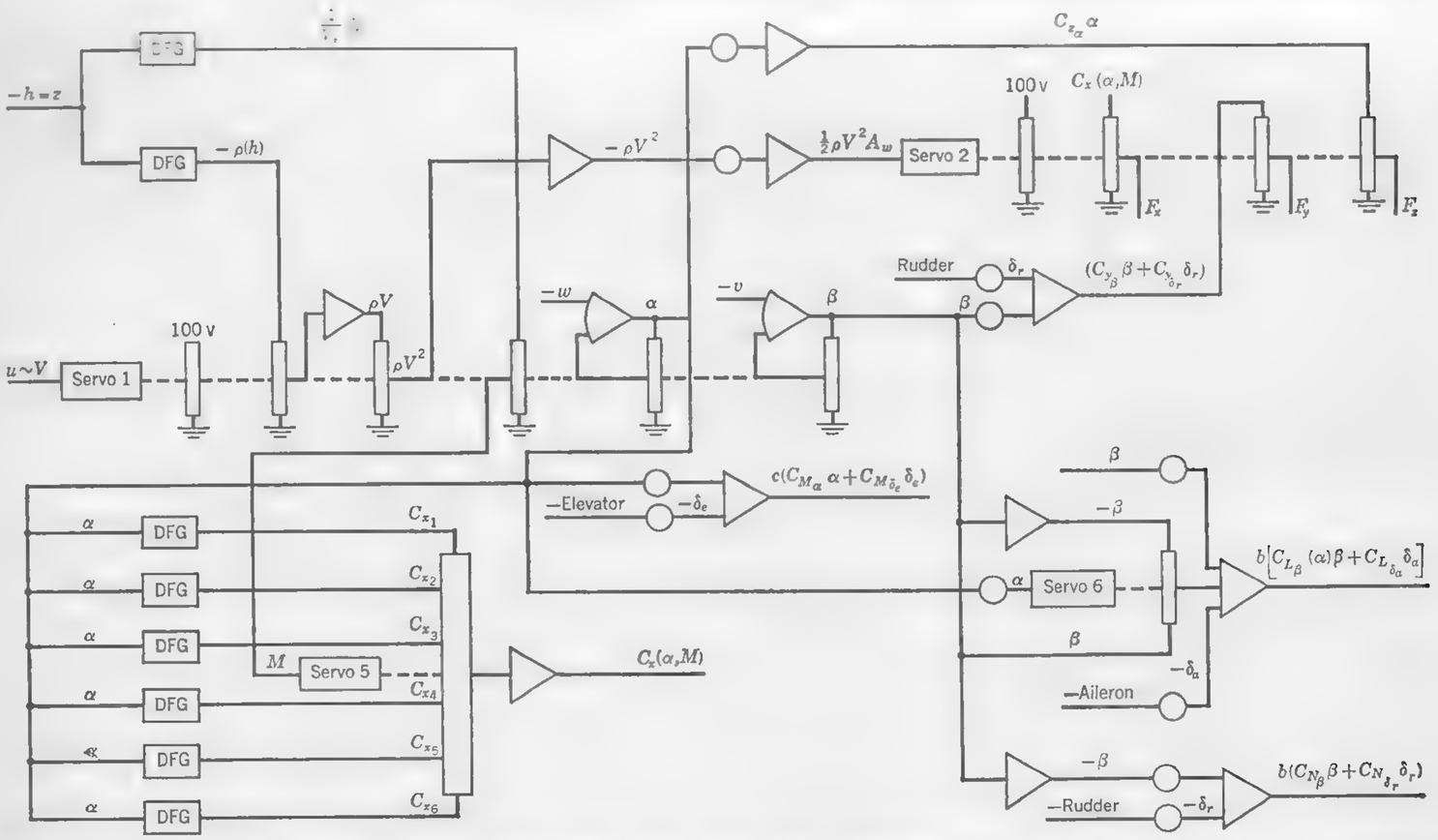


FIG. 5.5.3c. Computation of angles of attack, Mach number, and aerodynamic coefficients. The Mach number M is the result of multiplication of V by $1/V_s$, which is obtained as a function of the altitude h. A tapped potentiometer driven by servo 5 produces the drag coefficient C_x as a function of the two variables M and α (see also Sec. 3.3).

5.5.6. Typical Simplified Flight Equations. (*a*) *Longitudinal Analysis.* For a two-dimensional longitudinal study the use of wind axes is advantageous because of the simple expression which then results for α.

$$m\dot{V} = F_X' + T - mg\sin\gamma \tag{5.5.1b}$$
$$-mV\dot{\gamma} = F_Z' - T\alpha + mg\cos\gamma \tag{5.5.3b}$$
$$I_\gamma\dot{q} = M^s + M^d \tag{5.5.5b}$$
$$\dot{x} = V\cos\gamma \tag{5.5.7b}$$
$$\dot{z} = -V\sin\gamma = -\dot{h} \tag{5.5.9b}$$
$$\dot{\theta} = q \tag{5.5.11b}$$
$$F_X' = \tfrac{1}{2}\rho V^2 A_w C_X' \tag{5.5.13b}$$
$$F_Z' = \tfrac{1}{2}\rho V^2 A_w C_Z' \tag{5.5.15b}$$
$$M^s = \tfrac{1}{2}\rho V^2 A_w c(C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e) \qquad M^d = \rho V A_w \frac{c^2}{4} C_{M_q} q \tag{5.5.17b}$$
$$\alpha = \theta - \gamma \tag{5.5.19b}$$
$$M = \frac{V}{V_s} \tag{5.5.21b}$$
$$V_s = V_s(h) \tag{5.5.23b}$$
$$\rho = \rho(h) \tag{5.5.24b}$$

The last two may be omitted if the effect of variations in altitude upon the velocity of sound and the air density can be neglected.

(*b*) *Linearization.* If, further, the velocity is assumed constant, the ρV and $\frac{1}{2}\rho V^2$ servos needed in the generation of the aerodynamic forces and moments may be eliminated. Furthermore, the aerodynamic derivatives C_{Z_α}, C_{M_α}, and C_{M_q} may just as well be assumed constant. If γ remains small throughout, the system reduces to a system of linear differential equations of the third order, containing one equation each for $\dot{\gamma}$, $\dot{q}$, and $\dot{\theta}$.

$$\dot{\gamma} = C_1\alpha - C_2$$
$$\dot{q} = \ddot{\theta} = C_3\alpha + C_4 q$$
$$\alpha = \theta - \gamma$$

where C_1, C_2, C_3, C_4 are constants as follows:

C_1 lift coefficient + thrust
C_2 influence of gravity
C_3 static moment coefficient
C_4 dynamic moment coefficient

An examination of the equation for $\dot{q} = \ddot{\theta}$ shows that equation to contain one restoring term involving α, and two damping terms, one involving q directly, and one through the α term in the $\dot{\gamma}$ equation.

(*c*) *Longitudinal Stability.* The natural frequency and damping of this linear system can readily be determined, and they serve as a useful check on the computer solution. The longitudinal response of a conventional airplane to a small disturbance (elevator deflection, or a gust) actually corresponds to a damped sine wave. The static moment coefficient determines the restoring moment, and thus the period, which may be of the order of magnitude of a second or longer. In order for an aircraft to be statically stable this moment coefficient must have the proper sign. The damping, and thus dynamic stability, is largely determined by the dynamic moment coefficient.

The newer and faster planes tend to have smaller margins of stability, especially at the low speeds needed at take-off and landing; many computer problems deal with the action of *control systems designed to add stability.* In some missile-launching problems even static instability is encountered; then the problem is to determine whether the missile can be accelerated quickly enough to regain static stability at higher speeds.

(*d*) *Phugoid Oscillation.* In addition to the oscillation involved in the longitudinal stability of the aircraft, another kind of oscillation may occur; this one involves the velocity and the altitude. Suppose the airplane is climbing; then the velocity V will

tend to decrease [$mg\sin\gamma$ term in Eq. (1*b*)] and will decrease if the thrust is not increased. A reduction in V will cause a corresponding reduction in the lift force (α term in F_Z'); this will in turn tend to make $\dot{\gamma}$ negative, causing γ to decrease until α becomes negative. As this is taking place $\dot{V}$ becomes positive, and V becomes larger again (*phugoid oscillations*). These oscillations have a rather long period, usually measured in minutes. The minus sign in Eq. (3*b*) is necessary because of the sign convention used; a positive z direction is directed downward, while a positive angle of climb is directed upward.

(*e*) *Lateral Stability.* The lateral stability of the airplane can be traced through the equations of motion in a similar manner. Here the important coefficients are the yawing moment due to the angle of attack in yaw (*sideslip*), and damping moments due to yaw rate. In the equations for lateral motion, the introduction of the roll equation is of course very important. In a so-called coordinated turn, the roll angle is such that the g's pulled by the airplane lie in its symmetry plane. In the equation for the roll of the airplane it should be noted that there is no restoring moment due to angle of roll [no ϕ terms in Eq. (16)]. Roll stabilization is therefore obtained in conjunction with the yawing motion. The period of the combined motion (oscillation) in yaw and roll (*Dutch roll*) may be of the order of magnitude of one or more seconds. Some missiles are not stabilized in roll attitude, but only stabilized in roll rate. In these cases it is necessary to specify whether p or ϕ is to be kept zero. In order to achieve roll stabilization, it is necessary to introduce a control system operating from a gyro generating the roll angle.

(*f*) *Simplified Representation of Dynamics.* Frequently it is desired to study a complex guidance and control system for an aircraft without simulating the dynamic equations of the aircraft in any such detail as represented by Eqs. (1) to (24). Then the aircraft dynamics are simply represented by one or more *second-order transfer functions,* corresponding to the stability characteristics in pitch and yaw. In other words, the control system will put out certain signals, which after suitable second-order lags produce the corresponding displacements in pitch and yaw.

5.5.7. Limitations. Use of Direction Cosines. A potentially serious limitation in the use of Eqs. (1) to (12) is implied in the use of the successive rotations ϕ, θ, and ψ in order to define the body axes, and the use of ξ and γ to define the velocity vector. Whenever the second rotation (the pitch angle θ, or the angle of climb γ) becomes equal to 90 deg, the equations become singular, and therefore useless for computations involving ϕ and ψ and ξ. This happens in cases of vertical launching (VTOL aircraft, missiles) or vertical impact of missiles.

Two alternate methods of dealing with this difficulty are possible. First, the coordinate systems and the orientation angles may be redefined in such a way that the angle chosen for the second rotation no longer reaches 90 deg, and the appropriate changes may be made in the equations involving these angles. A second possibility is the avoidance of the use of angles in the definition of the body axes by the use of *direction cosines.* Let the nine direction cosines be denoted by l_i, m_i, and n_i, $i = 1, 2, 3$. Equations (10), (11), and (12) are now replaced by the following nine equations:

$$\left.\begin{aligned} \dot{l}_i &= m_i r - n_i q \\ \dot{m}_i &= n_i p - l_i r \\ \dot{n}_i &= l_i q - m_i p \end{aligned}\right\} (i = 1, 2, 3) \tag{5.5.25}$$

Of these nine equations only three are independent. The following six constraints have to be satisfied by the nine direction cosines at all times:

$$\sum_{i=1}^{3} l_i^2 - 1 = \sum_{i=1}^{3} m_i^2 - 1 = \sum_{i=1}^{3} n_i^2 - 1 = 0$$
$$\sum_{i=1}^{3} l_i m_i = \sum_{i=1}^{3} m_i n_i = \sum_{i=1}^{3} n_i l_i = 0 \tag{5.5.26}$$

If the nine differential equations (25) are solved with perfect accuracy subject to the correct initial conditions, the six constraints will be satisfied automatically. In an actual problem setup, however, this situation may not prevail. One method to deal with this difficulty is to compute error voltages equal to the differences between the expressions (26) and zero, and to use these errors strategically to correct the computed values of l_i, m_i, and n_i. The logical place to apply these corrections is in the integrators generating the direction cosines. It is likely, however, that such a correction scheme will involve a prohibitive amount of equipment. A possible compromise is to compute the nine direction cosines open-loop from Eq. (25), to keep track of the errors by computing the expressions (26), and to hope for the best.

The direction cosines are related to the angles ϕ, θ, and ψ as follows:

$$\begin{aligned}
l_1 &= \cos\psi\cos\theta \\
l_2 &= -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi \\
l_3 &= \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\
m_1 &= \sin\psi\cos\theta \\
m_2 &= \cos\psi\cos\phi + \sin\psi\sin\phi\sin\theta \\
m_3 &= -\cos\psi\sin\phi + \sin\psi\cos\phi\sin\theta \\
n_1 &= -\sin\theta \\
n_2 &= \cos\theta\sin\phi \\
n_3 &= \cos\theta\cos\phi
\end{aligned}$$

CONTROL SYSTEMS

5.5.8. Airplanes. The simplest aircraft-control systems are low-cost autopilot systems intended to maintain aircraft altitude or heading only, and dampers designed to improve dynamic stability. On the other hand, one may have very complex systems which compute, say, the proper course for an interceptor and then apply appropriate signals to the control surfaces to achieve this course; the system may also present suitable displays to a human pilot who controls the aircraft or monitors the system.

5.5.9. Missiles. Missile-guidance systems assume many different forms depending upon the mission and, sometimes, upon the stage of the flight path. *Command-guidance systems* are controlled by a control station on the ground, on a ship, or in an airplane. *Beam-rider systems* cause the missile to follow a radar beam from a control point, and *homing systems* cause the missile to home on the target. Ground-to-ground missile guidance may be also *preprogrammed* or employ *inertial navigation* (integration of accelerations measured on a stable platform in the missile) or *celestial navigation*.

5.5.10. General Procedures. The control system is usually described by a block diagram showing the operations to be performed in each block in terms of their transfer characteristics (see also Sec. 5.3). The block diagram must then be put into mathematical form, with precisely defined input variables obtained from the aircraft kinematics. All quantities must be expressed in consistent units. The mathematical relations are then programmed on the analog computer, which produces voltages representing the control-surface deflections. These voltages, in turn, are used as inputs to the computing-element blocks implementing the aerodynamic equations (Fig. 5.5.1); care is necessary to ensure correct signs for all variables in a large computer setup.

5.5.11. Some Special Techniques. Two serious problems frequently encountered in trajectory computations are (1) the accurate generation of small differences between two large quantities and (2) accurate representation of variables which vary over a wide range.

(*a*) *Differences between Two Large Quantities.* As a typical example, consider the following situation encountered in a beam-rider problem. Referring to Fig. 5.5.4, the error distance ϵ between missile and beam, which is small compared with the missile coordinates X and Y, is given by

$$\epsilon = -X\sin\phi + Y\cos\phi$$

Unfortunately the two terms on the right are nearly equal, so that generation of ϵ tends to be inaccurate. This difficulty does not apply to the generation of

$$R = X\cos\phi + Y\sin\phi$$

It is also futile to generate ϵ in the form

$$\epsilon = R\left(\arctan\frac{Y}{X} - \phi\right)$$

since this again involves the generation of a small difference.

The difficulty can be overcome if the derivatives $\dot\phi$, $\dot X$, and $\dot Y$ are available in the computer setup. This will certainly be true for X and Y, and possibly also for ϕ.

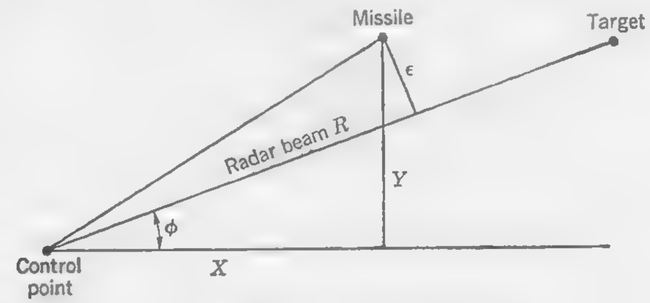


FIG. 5.5.4. Beam-rider geometry.

If ϕ is given only as arctan Y/X, one must generate $\dot\phi$ as a function of X, Y, $\dot X$, and $\dot Y$, obtained by differentiation of arctan Y/X. If, now, ϵ is generated by integration of

$$\begin{aligned}
\dot\epsilon &= -(X\cos\phi + Y\sin\phi)\dot\phi + (-\dot X\sin\phi + \dot Y\cos\phi) \\
&= -R\dot\phi - \dot X\sin\phi + \dot Y\cos\phi
\end{aligned}$$

then the necessity of taking the small difference of two large quantities no longer exists; in fact, ϵ may change at a rate comparable with the rates $\dot X$, $\dot Y$, and $\dot\phi$. In addition, ϵ is now an integrator output and can be scaled independently of X and Y, so that accuracy may be further improved.

This method, which is quite generally applicable, will require more equipment, especially if some of the terms making up the derivative of the desired difference must be generated; but the only alternative may be a computer solution so inaccurate as to be useless.

(*b*) *Variation of Variables over a Wide Range.* The accurate handling of variables which vary over a wide range is again best illustrated by an example, as in the following homing problem.

Let ϵ be an error angle, measured in the missile coordinate system, and X' and Z' the components of the relative distance from missile to target, also measured in the missile coordinate system (Fig. 5.5.5), and $\tan\epsilon = Z'/X'$; or, for small angles $\epsilon = Z'/X'$.

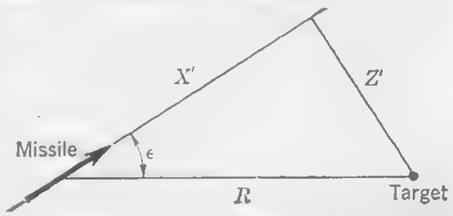


FIG. 5.5.5. Homing missile. The target is observed at distances X' and Z' in the missile coordinate system. The control system acts to reduce the angle ϵ to zero, causing the missile to home on the target.

The error angle is the main input to the control system; presumably the effect of large values of ϵ will be limited. It is desirable to calculate ϵ accurately, especially near the end of the trajectory, where correct action of the control system is particularly important. Note that X' will be quite large at the beginning of the problem—perhaps several hundred thousand feet, and Z' may or may not become large sometime during the running of the problem. Near the end, however, both X' and Z' will approach zero. The difficulty, then, consists of maintaining the accuracy in the generation of X' and Z' and ϵ throughout this large variation in range.

The difficulty is overcome through the use of a *variable scale factor* K for X and Z. The machine variables are now KX' and KZ', and

$$\epsilon = \frac{KZ'}{KX'}$$

as before.

The variable scale factor K may be generated in different ways. Sometimes it may be possible to use a step function or a limited ramp function for K, but usually this introduces undesirable transients into the problem. The following method, which computes K based on the instantaneous value X', has been found the most satisfactory one.

In Fig. 5.5.6, K is generated by an integrator whose input is an "error" equal to the difference between KX' and a fixed voltage F. Since K is the multiplying factor for X', this circuit serves to maintain KX' equal to F as long as it is physically possible.

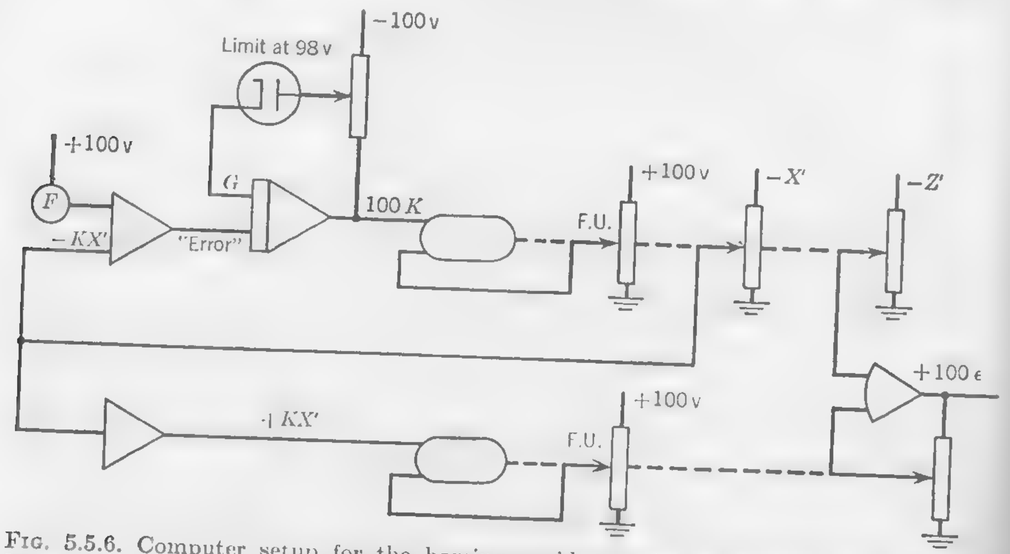


FIG. 5.5.6. Computer setup for the homing problem, using a variable scale factor K. When X' is large (at the beginning of the problem) the scale factor K multiplying X' and Z' is small. As X' decreases, K increases (until it reaches the limit imposed by the maximum excursion of the K servo). Thus a larger scale factor K is used for smaller values of X' and Z' for greater accuracy in the computation of ϵ.

Naturally, K has a certain upper limit dictated by the maximum excursion permitted by the computer. However, K may easily vary through a range of 1 to 100, and what is most important, the generation of KX' and KZ' will be most accurate when K is largest, i.e., near the end of the trajectory when X' becomes small.

The square of the miss distance, $X'^2 + Z'^2$, is easily computed and recorded. Since the initial large values of $X'^2 + Y'^2$ are not of interest, one may use a large scale factor for this quantity and limit X' and Y' in the miss-distance computation. A record of $X'^2 + Y'^2$ will show it to be at some limited value most of the time and will show a decrease to a minimum, and an increase at the end of the trajectory. The minimum can be read easily and accurately; using a medium-accuracy computer with 0.05 per cent passive components and 0.025 per cent servo-potentiometer linearity, miss distances have been calculated with an accuracy of 1 part in 20,000. More accurate computers with better dynamic performance could further improve these results.

NON-RIGID-BODY MOTION

5.5.12. Treatment of Aeroelasticity.[7 to 10] The preceding discussion has been limited to a treatment of the aircraft as a rigid body. The problem becomes considerably more

complicated when deformations of the airplane structure are to be included as well. The problem of aeroelastic deformations is generally treated separately from the rigid-body dynamics of the aircraft, but at times interactions of some aeroelastic oscillation modes and control-system behavior must be studied.

The equations describing the structural deformations are partial differential equations; methods of solving such problems on electronic analog computers are treated in Sec. 5.10, and network-computer methods are described in Secs. 9.1 and 9.2. In particular, relatively simple beam analogies (Sec. 5.10) can describe the motion of an aircraft wing of large aspect ratio, either straight or swept back. Fuel tanks and engine pods may be added at will. Wings of small aspect ratios, such as delta wings, must be described by two-dimensional (plate) analogies.

SOLUTION OF LARGE PROBLEMS

5.5.13. Special Precautions. The precautions and techniques needed to deal with large problems are essentially an extension of those used with smaller problems.[11] The patching procedure is the same as for smaller problems; automatic checking procedures (Sec. 4.3) for the correctness of the patching are naturally more important now.

Problems may become large for different reasons: it may be that many complicated effects must be included in a six-degree-of-freedom study of an aircraft or a submarine; this will—especially in the aerodynamic computations—require a great deal of equipment. It may also happen that a relatively simple set of equations involves a large number of different dependent variables, e.g., in structural or process problems requiring solution of partial differential equations.

Dynamic checks are best carried out by testing (1) linearized portions of the problem and (2) decoupled portions of six-degree-of-freedom problems which have only one or two degrees of freedom and, in the case of structural problems, by testing (3) the computing loops corresponding to the motion at a single station.

The dynamic checks carried out for such subsystems can frequently be checked analytically, or their expected behavior may be known from previous analyses. As the large problem is then connected up, the resulting effects should be studied carefully and examined as regards their reasonableness. By following these steps carefully, one may obtain the solution to large problems with considerable degree of confidence in the results.

Since no system of partial checks can take the place of a check solution to the whole problem, it will be useful to obtain a digital-computer check solution for a few special cases, and to rely upon continuity considerations as parameters are changed.

The importance of good bookkeeping practices also becomes more crucial when the problems become larger. This means that all schematics, flow diagrams, and setup sheets must be made up carefully, with all interconnecting trunk lines ("inters") and multiples shown and labeled. Whenever the setup is changed—no matter how temporary the nature of such a change—the corresponding changes must be made on the schematics without delay, and also on all tables where information regarding the interconnections is recorded. Simple changes, which could easily be remembered in small problems, cannot be entrusted to the operator's memory in large problems.

With sufficient care, analog-computer solution of complex problems involving well over 250 amplifiers is feasible and practical.

References

Most of the aerodynamic background material can be found in W. F. Durand, *Aerodynamic Theory*, 6 volumes. Additional material with emphasis on guided missiles may be found in the series on *Principles of Guided Missile Design*, Grayson Merrill, Editor, in particular in the volume *Guidance*, by Arthur S. Locke, D. Van Nostrand Company, Inc., 1955.

Many additional papers on the use of analog computers in this field have been presented. The following is only intended to be a partial list, and many added references will be found in the individual papers.

1. REAC Techniques, *Project Cyclone Symposium I*, Reeves Instrument Corporation and U.S. Navy Special Devices Center, March, 1951.
2. Simulation and Computing Techniques, *Project Cyclone Symposium II*, Reeves Instrument Corporation and U.S. Navy Bureau of Aeronautics, May, 1952.
3. Simulation and Computing Techniques, *Project Typhoon Symposium III*, U.S. Navy Bureau of Aeronautics and Naval Air Development Center, October, 1953.
4. National Simulation Conference, January, 1956, Dallas, Tex.
5. *Proc. First Flight Simulation Symposium*, White Sands Proving Grounds, New Mexico, November, 1956.
6. *Proc. Natl. Simulation Council*, 1957 National Meeting, December, 1957, Washington, D.C.

The proceedings of the various branches of the Simulation Councils, Inc., which appear in *Instruments and Control Systems* (formerly *Instruments and Automation*) contain discussions of many related problems.

The proceedings of the Joint Computer Conferences (IRE-AIEE-ACM) which are held twice a year (once on the East Coast and once on the West Coast) contain material of interest, although greater emphasis is on digital techniques rather than analog.

Papers on electronic-analog-computer techniques for aeroelastic problems include the following:

7. Winson, J.: The Solution of Aeroelastic Problems by Electronic Analog Computation, *J. Aeronaut. Sci.*, July, 1950.
8. Baird, E. F., S. Pines, and J. Winson: *Project Cyclone Report on a Flutter Problem*, Reeves Instrument Corporation and U.S. Navy Special Devices Center, February, 1952.
9. Yanowitch, M.: *Project Cyclone Solution of a Flutter Problem*, Reeves Instrument Corporation and U.S. Navy Bureau of Aeronautics, June, 1955.
10. Clymer, A. B.: *Trans. IRE/PGEC*, September, 1959.

Large-problem solutions are discussed in

11. Karen, A., and B. D. Loveman: Large Problem Solutions at Project Cyclone, *Instr. and Automation*, January, 1956.

PART 6: PARTIAL SYSTEM TESTS AND FLIGHT TABLES

By LOUIS BAUER

INTRODUCTION

5.6.1. Partial System Tests. A *partial system test* is a test involving actual components of a physical system under investigation, with a computer operating on a one-to-one time scale substituting for part of the system or its environment. (Definition by IRE subcommittee 8.8.) Partial system tests (or *physical simulation*, or *test simulation*) involve some of the most important application of analog computers. In any development of new equipment—whether aircraft, aircraft components, autopilots, or guidance systems—the time arrives when a finished component has been built and must be tested under realistic conditions before it is actually used. No matter how thorough the theoretical analysis has been, the use of a mathematical model is basically limited by the fact that exact mathematical relationships may not be known accurately. Realistic results can then be produced only by the use of actual equipment.

Referring back to a typical missile-system block diagram (Fig. 5.5.1), partial system tests can substitute hardware for any of the blocks shown there. In particular the guidance and control equipment is most frequently tested by such a procedure; this part of the system constitutes a part of the loop for which it is difficult to find an exact mathematical representation.

PARTIAL SYSTEM TESTS INCLUDING A HUMAN OPERATOR

5.6.2. Flight Simulators. Partial system tests may be classified according to whether or not the equipment tested includes a human pilot. When it does, the computers so used are frequently employed as *training devices;* or the control system itself may also be tested to determine whether it can be operated by a human pilot, and to find optimum parameters for such operation.

Training devices take many different forms depending upon the type of training to which the pilot or crew is to be subjected. An *operational flight trainer* (OFT) includes the simulated cockpit of an airplane together with all necessary controls to permit the pilot to learn to fly the airplane. Signals applied to the controls by the pilot are fed to the computer, which in turn determines the behavior of the airplane in response to such signals. The computed aircraft behavior is translated into signals observed by the trainee on instruments and, possibly, also sensed as cockpit motion. It is possible to go to considerable lengths in order to present a realistic "picture" to the pilot. If only instrument deflections are to be produced, the job is relatively simple. On the other hand, one may produce the illusion of motion by enclosing the cockpit, e.g., by projecting moving pictures of the surrounding scenery on the walls within the pilot's range of vision. Aircraft roll, pitch, and yaw can be simulated by suitable motions of the visible horizon. The effects of acceleration can also be simulated by straps holding the pilot in his seat with adjustable and controllable forces. Another more realistic way of subjecting the pilot to g forces involves the use of a centrifuge; this is, of course, not compatible with the use of motion-picture displays.

5.6.3. Missile-guidance Trainers. The operation of a command-guidance system for a guided missile is conveniently learned on a simulator. Here again the dynamic response of the missile to the control signals is computed on the analog computer, and a suitable display (plotting board, cathode-ray oscilloscope, or radar screen) is presented to the operator; he in turn operates the control system which generates the control-surface deflections necessary to guide the missile. The value of such a procedure is obvious; not only can operators be trained efficiently without expenditure of actual missiles, but they may also be presented with emergencies in the form of malfunctions of missile components, incorrect action of the control system, etc. If a missile is to be launched from an airplane, the training of the pilot can be accomplished in two stages: first, the correct control procedure is learned on the ground; next, a small computer is programmed, set up in the airplane, and taken aloft. Firings and control of air-borne missiles can thus be learned and practiced with additional realism, but still without expenditure of actual weapons.

5.6.4. Other Training Devices. Training devices are also frequently used when the operation to be learned does not involve an aircraft: in this class are submarine simulators and nuclear-reactor simulators (Sec. 5.8). These are used for instruction of personnel who will eventually operate submarines, nuclear reactors, etc.

COMPONENT AND SUBSYSTEM TESTS

5.6.5. Flight Tables. Similar techniques apply when the primary purpose of partial system tests is the testing of hardware rather than the training of human operators. In testing guidance equipment containing gyros or similar devices, *flight tables* and *roll tables* are used extensively. A flight table is a platform which can be subjected to the same rotations as the aircraft (plane or missile) under investigation. Air-borne equipment is rigidly mounted on this platform and subjected to the same angular accelerations, velocities, and displacements as would be produced in free flight. The control signals put out by this equipment in turn serve as inputs to an analog computer (flight simulator) which computes the resulting aircraft motion and positions the flight table accordingly. As a general rule of thumb, a high-quality flight table can position its pay load with the same static and dynamic accuracy as an analog-computer resolver servo (Sec. 3.1).

Depending upon the degree of realism required in the simulation and the complexity of the control system under test, flight tables may be built for one-, or two-, or three-degree-of-freedom studies (*one-, or two-, or three-axis flight tables*). Because of cross-coupling effects one cannot, in general, use two one-axis tables instead of one two-axis table. Flight tables may be positioned according to position signals (ϕ, θ, ψ) or rate signals (p, q, r) or acceleration signals. Since the angular-velocity components p, q,

and r are mutually perpendicular, the axes of a two- or three-axis flight table must be gimbaled so as to permit two or three independent mutually perpendicular rotations. The innermost gimbal drives the platform carrying the pay load; going out from the innermost gimbal, the other gimbals have to drive not only the pay load but also the inner gimbals and their drives. If any of the gimbals has provision for continuous revolution, it will be necessary to provide slip rings to carry power to the inner gimbals. The complications (and expense) involved in constructing two-axis flight tables thus are considerably greater than for single-axis tables; a three-axis table, in turn, is much more complex in construction and operation than a two-axis flight table.

5.6.6. Roll Tables (Single-axis Flight Tables). Roll tables are used primarily because of their comparatively low cost and simplicity of operation. Commercial models are available from many manufacturers.

In principle, the operation of roll tables is entirely analogous to the operation of two- or three-axis flight tables. The components to be tested are mounted on a platform which is driven by a servo. The servomotor, which must be sufficiently powerful to produce the required accelerations, may be used as a position servo, a rate servo, or as an acceleration servo; i.e., the commands to the servo may be signals corresponding to an angular displacement, angular rate, or angular acceleration.

If the roll table is driven by a *position servo* with potentiometer follow-up, d-c analog-computing signals may be used directly as the input; in this case, the range of rotation is limited, and mechanical limit stops must be provided. A position servo may also be used with a synchro (control transformer) follow-up. In this case, the rotation need not be limited, and the input is a synchro signal; but such signals may not be readily available from d-c analog computers.

It is more likely that the roll table will be connected as a *rate servo*. In this case, the feedback may be from a tachometer which detects the rate of rotation of the shaft; the angular displacement which results will therefore be the integral of the input. Accurate d-c tachometers are not more expensive than accurate a-c tachometers. When tachometers are used, care must be taken to preserve the servo loop gain at low rates.

The rate of rotation of the roll table can also be accurately controlled by a rate gyro or integrating gyro. It is then possible to use d-c analog-computing signals directly as the inputs to the servo.

For very sensitive applications it is necessary to observe the distinction between the rate of shaft rotation and the rate of rotation of the platform, since the latter includes the effect of the earth's rotation.

Roll tables are usually provided with a self-contained power supply, and d-c to a-c modulators where necessary. For optimum operation over different ranges of displacements or rates, different gear ratios between motor and platform can be selected by means of a gear-shifting scheme. While generally not designed to drive heavy loads, roll tables are frequently useful in the testing of components and small subassemblies. By virtue of their small size they do not require special handling equipment and do not pose any particular installation problems. Leveling adjustments are easily made.

5.6.7. Three-axis Flight Tables. For the most realistic simulation a three-axis flight table will be required. At the same time, three-dimensional simulation will involve larger and heavier components under test. The innermost gimbal will therefore have to have the capability of driving rather heavy loads at the high angular velocities and accelerations often encountered in modern missiles. This makes the use of powerful servos mandatory, and since they in turn will add to the mass of the inner gimbals, the outer gimbal drives will have to be correspondingly more powerful and massive. Amplidynes and hydraulic servos are commonly employed in these applications.

One limitation in the use of a three-axis table is *gimbal lock;* the excursion of the middle gimbal has to be kept below 90 deg to avoid gimbal lock. For complete freedom of angular motion four gimbals are needed; the motion of the fourth gimbal must be controlled so that gimbal lock is prevented. This may involve a control system which will cause a corrective motion of the fourth gimbal whenever the displacement of the gimbal causing the difficulty approaches one of its limits.

Different design philosophies have resulted in a considerable range of flight-table sizes. On one hand, a flight table can be designed for general-purpose applications and therefore with the best possible capability of driving large loads at high accelerations with small errors. This results in very large, massive units with correspondingly powerful servos. On the other hand, the development of a flight table *for a specific application* can result in substantial savings, since the accelerations, velocities, and displacements actually encountered may be far below those for which general-purpose flight tables are designed.

5.6.8. "Gimbal Computation" in Flight Tables. Gyro-controlled Flight Tables. A number of methods for connecting flight tables into a flight-simulator loop will be described next. If a complete flight simulator is already available, one can simply "slave" the flight-table gimbals to the appropriate Euler-angle outputs of the flight simulator (*slave operation*, Art. 5.6.10). It may, however, be more economical, and

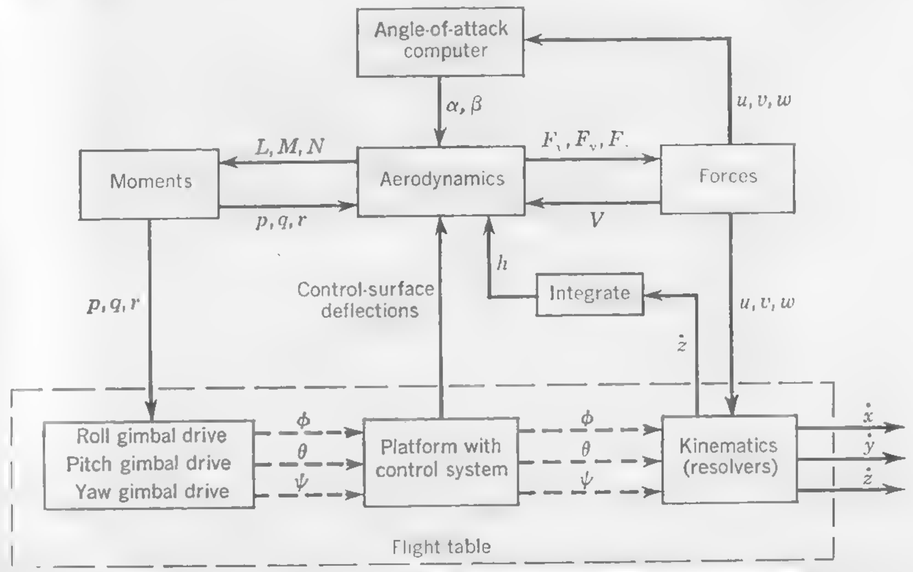


Fig. 5.6.1. Flight-table block diagram. The inputs to the flight table are the angular rates p, q, and r which are computed from the moment equations. The control system located on the flight table puts out the appropriate control signals. Other computing components requiring the angles ϕ, θ, and ψ as inputs must be located on the flight table and driven by the roll, pitch, and yaw servo, respectively. The symbols are those of Table 5.5.1.

presumably more accurate, to make the flight table an integral part of the computer setup by controlling the gimbal-servo *accelerations* or *velocities*, so that *integrations and angular resolutions are performed by the flight table itself* (*gimbal computation*).

In the block diagram of Fig. 5.6.1, the dashed lines correspond to mechanical connections; the control system is rigidly mounted on the innermost gimbal, which assumes the orientation angles ϕ, θ, and ψ in space (see also Sec. 5.5). One method commonly employed uses *rate gyros or integrating gyros* to detect p, q, and r (Fig. 5.6.2). The inputs to the flight table are the desired angular rates p, q, and r, which are computed from the moment equations (5.5.4) to (5.5.6). Three gyros mounted on the innermost gimbal detect the actual rates achieved by this gimbal. These rates are compared with the desired rates p, q, and r to produce errors used to drive the roll, pitch, and yaw servos so that the displacements of these gimbals will be equal to the angles ϕ, θ, and ψ. In this fashion the rotational equations of motion (5.5.10) to (5.5.12), viz.,

$$\dot{\phi} = p + (q \sin \phi + r \cos \phi) \tan \theta$$
$$\dot{\theta} = q \cos \phi - r \sin \phi$$
$$\dot{\psi} = (q \sin \phi + r \cos \phi) \sec \theta$$

are solved without any resolvers if the angular displacements remain small. The flight-table gimbals produce rotations in the order *yaw-pitch-roll*.

If the angular displacements become large (e.g., large roll angles), then a resolver for error resolution must be added to permit the errors in pitch and yaw rate to produce the proper corrective actions in the respective servos, and to keep the servo gain constant. If the pitch or yaw angles build up to values much above 30 deg a second resolver may be necessary. One resolver is shown in Fig. 5.6.2.

The integrations required in the solution of these equations are accomplished by the rate gyros or integrating gyros in the feedback loops of the roll, pitch, and yaw gimbals. The trigonometric relations in the equations are satisfied by virtue of the geometric relationship between the innermost gimbal (where the rates p, q, and r are measured)

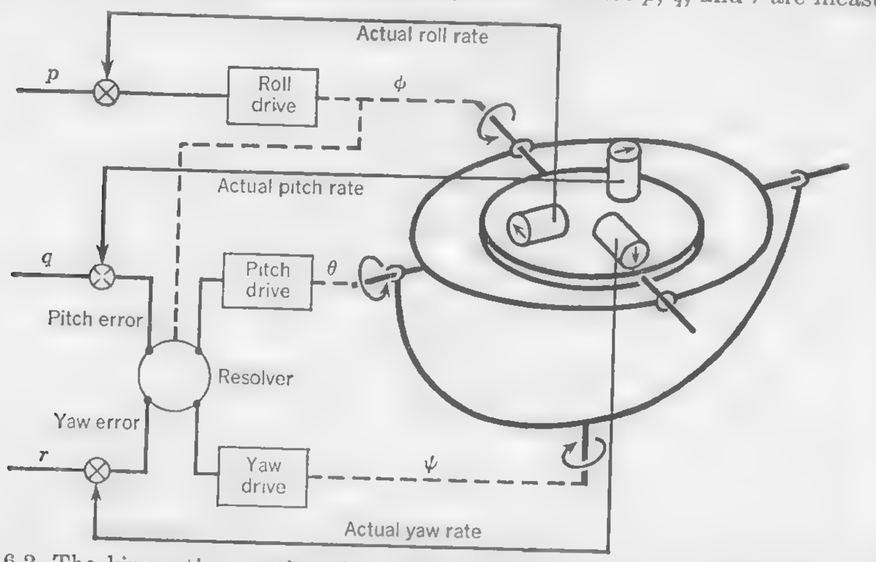


FIG. 5.6.2. The kinematic equations for rotation (yielding ϕ, θ, and ψ from p, q, and r) are solved by using three rate gyros or integrating gyros mounted on the flight table with their three axes mutually perpendicular. The excursion of the middle gimbal (pitch in this case) must be limited to prevent gimbal lock. A resolver which permits continuous roll rotation is shown; this resolver apportions the other gyro error signals appropriately to the pitch and yaw servo motors. A second resolver may be necessary for either pitch or yaw if the angles are allowed to build up much above 30 deg. This will be necessary to keep the servo loop gain constant.

and the drives for the three gimbals which generate ϕ, θ, and ψ. Commercially available integrating gyros permit good accuracy. If other computations and resolutions involving ϕ, θ, and ψ have to be carried out (as, for instance, in the kinematic equations), the necessary computing components (resolvers, potentiometers) are mounted so that they can be driven by the corresponding gimbal servos.

5.6.9. Tachometer-controlled Flight Tables. The following method uses *tachometers* to detect the rates $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ and is the one used for example, in the Bendix flight table. This flight table produces the rotations in the order *pitch-yaw-roll;* in the equations implemented by this flight table, ϑ and ψ and q and r are interchanged and the signs of ϑ and q are inverted:

$$\dot{\phi} = p - \tan\psi\,(q\cos\phi - r\sin\phi)$$
$$\dot{\theta} = \sec\psi\,(q\cos\phi - r\sin\phi)$$
$$\dot{\psi} = q\sin\phi + r\cos\phi$$

The block diagram is still that of Fig. 5.6.1, because the inputs to the flight table are still the rates p, q, and r. However, the angles ϕ, θ, and ψ are obtained by a more explicit solution of the equations of motion. The inputs to the pitch, yaw, and roll gimbals are the rates $\dot{\theta}$, $\dot{\psi}$, and $\dot{\phi}$; the desired values of these rates are computed from q, r, and p by means of resolvers and a secant potentiometer mounted on the roll and

yaw servo. These rates are matched against the servo rates determined from tachometers mounted on the three shafts. The inner gimbal will again be driven to the correct position determined by ϕ, θ, and ψ; 400- or 1,000-cps a-c computing signals are used in this application, and accurate a-c tachometers and resolvers are obtainable commercially (Fig. 5.6.3).

Note that the resolvers shown in Figs. 5.6.2 and 5.6.3 are a-c (induction) resolvers, and not sine-cosine potentiometers. The two outputs from a resolver with inputs q and r and angle ϕ are $(q\cos\phi - r\sin\phi)$ and $(q\sin\phi + r\cos\phi)$, respectively. Here, as before, additional resolvers and potentiometers are mounted on the three gimbal drives in order to permit computations involving the angles ϕ, θ, and ψ.

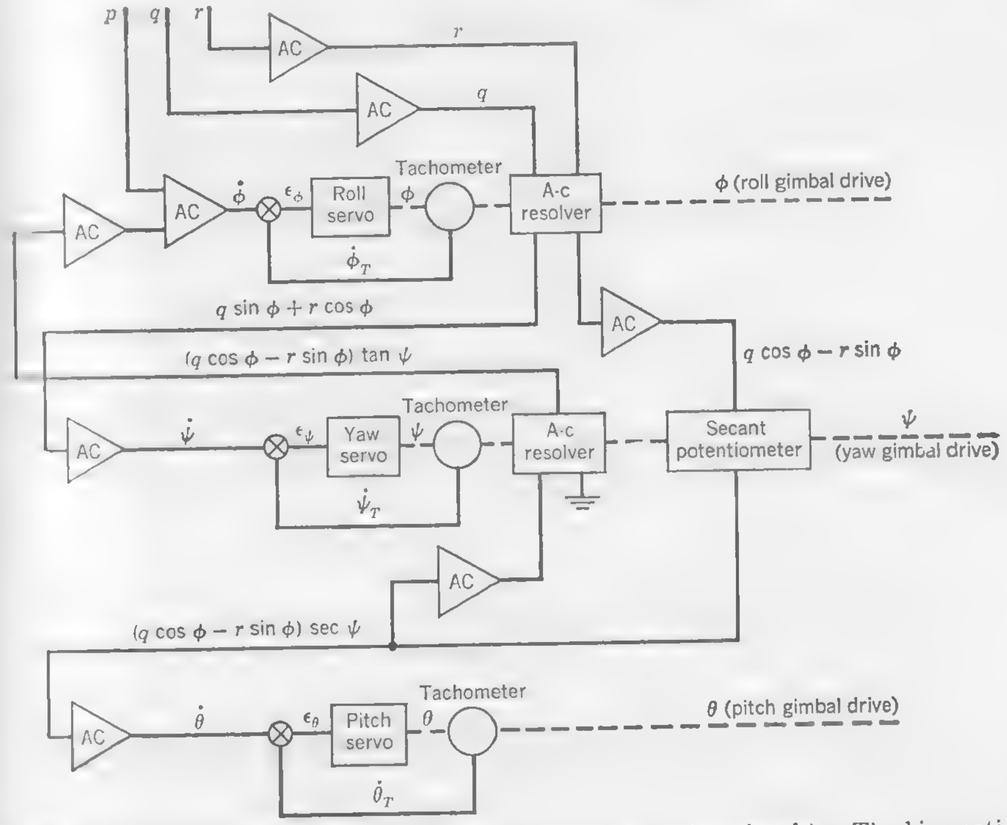


FIG. 5.6.3. Bendix flight table. The inputs are p, q, and r (a-c signals). The kinematic equations for rotation are solved using tachometers, resolvers, and a secant potentiometer. The desired (computed) rates $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ are compared with the actual rates measured by means of tachometers; the error signals so obtained drive the corresponding servos. The excursion of the middle gimbal (yaw in this case) must be limited to prevent gimbal lock.

5.6.10. Position-controlled Flight Tables. A third method uses ϕ, θ, and ψ as inputs to the flight table driven by *position servos*. The angles ϕ, θ, and ψ are obtained from p, q, and r in a separate attitude computer. The appropriate block diagram is shown in Fig. 5.6.4.

Other computations involving the angles ϕ, θ, and ψ are now accomplished with the aid of the signals from the attitude computer and suitable computing components. The advantage of this particular method is that no computing elements have to be mounted on the flight table itself.

If the position pickoffs are potentiometers, it is not possible to obtain unlimited rotation directly. If unlimited rotation is required the angle which is not limited has to be computed by means of an integrating servo instead of d-c integrator, and the angular position transmitted to the flight table by means of a synchro signal.

5.6.11. Installation. For the installation of a flight table, adequate planning has to take into account not only the sources of power which will be required—hydraulic as well as electric power, and a cooling system for this—but also the weight of the flight table, and its mobility. In this connection, the suggested layout for the Bendix flight table may be taken as typical. Here the flight table is set up in a room with direct access to the outside; it is then possible to conduct line-of-sight aircraft-tracking simulation with tracking equipment mounted on the flight table.

5.6.12. Other Partial System Tests. Many partial system tests do not require the use of flight tables. As before, the actual control system is substituted for the equations representing it. It may be that the action of the system does not depend

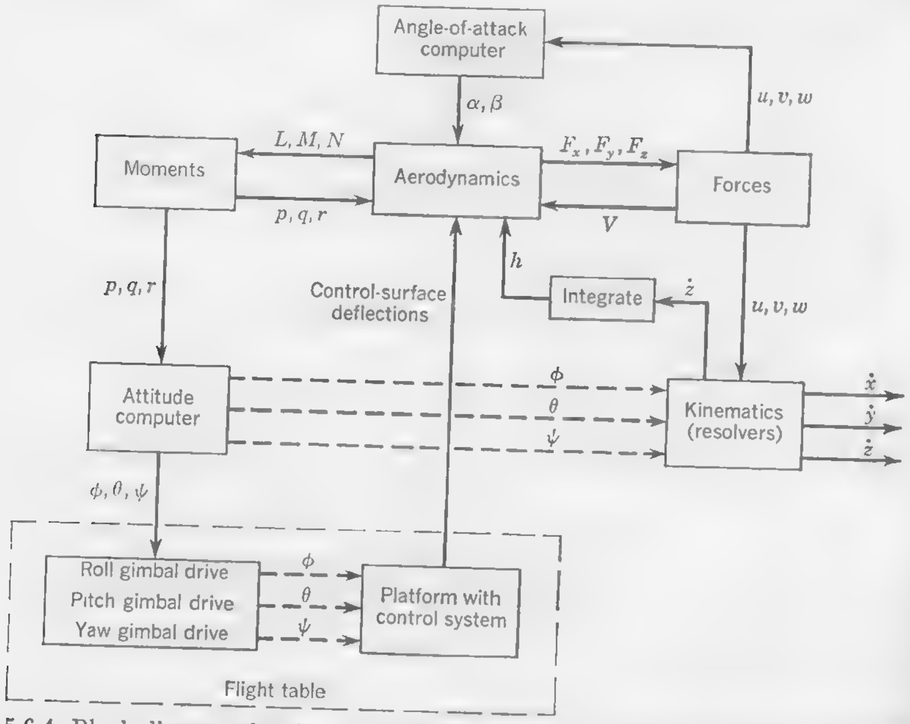


FIG. 5.6.4. Block diagram showing a possible setup for driving a flight table with position servos. The angles ϕ, θ, and ψ are computed in a separate attitude computer.

upon gyros, or that it need not be subjected to the same rotations as the missile. It may even be a remote-control system located on the ground. The simulation of the missile motion is now carried out on the computer, and information regarding it is fed into the control system. This will then put out command signals which will—properly translated and transmitted—cause the desired change in the course of the missile; the resulting course of the missile is again computed on the analog computer.

5.6.13. Discussion. Test-simulation problems usually have as their objective (1) *determination of actual system performance,* as contrasted with theoretical performance predicted from mathematical models, and (2) *optimization of control parameters.* After mathematical analysis and simulation have revealed the optimum parameters, it may happen that these are not realizable. Then one wishes to ascertain the best configuration actually attainable.

Practice has shown, however, that these tests usually accomplish much more than their stated objective. Partial system tests are usually the first occasion when newly assembled components and systems are used. For that reason many unexpected difficulties in operation are discovered and eliminated for the first time. This aspect of test simulation is possibly the most useful, although it is rarely among the stated objectives. It is important to keep this in mind when scheduling a simulation facility for

use in partial systems tests. When a new piece of equipment is to be set up for testing it is a good idea to allow a generous amount of time for unscheduled trouble shooting.

References

Specifications for flight tables may be found in:

1. Bendix Aviation Corporation, Research Laboratories Division, Detroit 1, Mich., "3-dimensional Flight Systems Simulator," a description of the flight table and simulator with specifications.
2. California Technical Industries, Division of Textron, Inc., Belmont, Calif., catalogue describing Test Instruments and Systems, including specifications of a three-axis flight simulator.

References 1, 2, 4, 5, and 6 listed after Sec. 5.5 also contain papers on the same subject

The proceedings of the various branches of Simulation Councils, Inc., which appear in *Instruments and Control Systems* (formerly *Instruments and Automation*) contain discussions of many simulation procedures.

PART 7: PROCESS-CONTROL APPLICATIONS

By CHARLES W. WORLEY

INTRODUCTION

5.7.1. Process-control Systems. Figure 5.7.1 shows the components to be represented in a typical process-control problem. Note that *controllers* comprise equalization for the feedback loop (proportional, rate, reset control) and that transmission of variables between blocks may involve *transport lags* (Sec. 6.1). A typical *actuator* is

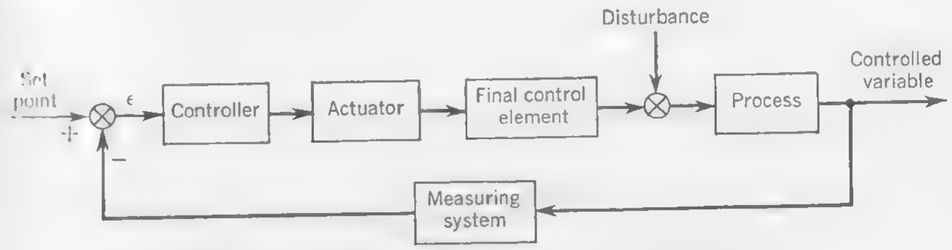


FIG. 5.7.1. Block diagram of a process-control system.

a motor or solenoid driving a valve as the *final control element.* The *objective of process-control design* is to create a stable system which minimizes some measures of deviation from a desired state or, more fundamentally, maximizes some payoff function. If the *error* $\epsilon(t)$ denotes the deviation of some process variable from its desired *set-point value,* one may wish to minimize max $|\epsilon|$, $\int_0^T |\epsilon| dt$, $\int_0^T \epsilon^2 dt$, or ask for *minimum overshoot* or *minimum cycling.* The resulting *parameter optimization problem* is different for different processes; history and theory of process-control techniques will be found in the literature.[1 to 16]

5.7.2. Process Models and Test Data. The choice of a specific model for a process and/or control system requires professional knowledge together with test data; the latter are often costly to obtain and difficult to interpret. References 19 and 20 deal with analog-computer (correlator) techniques for obtaining test data with a minimum of disturbance to existing processes. Where questionable data are the only ones obtainable, the importance of the parameter values in question must be checked by careful *parameter-variation studies,* which may lead to new specifications.

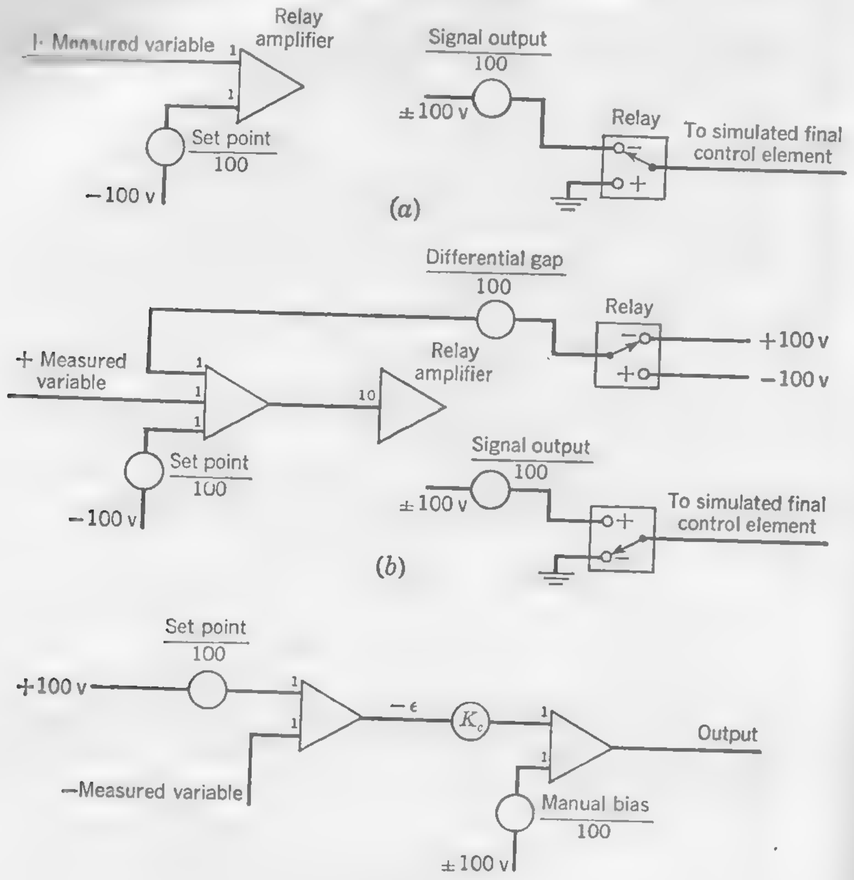


FIG. 5.7.2. Analog-computer representations of idealized industrial controllers: two-position controller (*a*), two-position controller with differential gap (*b*), proportional controller (*c*), and proportional-speed floating controller (*d*).

ANALOG-COMPUTER REPRESENTATION OF THE CONTROL SYSTEM

5.7.3. Linearized Representation of Control-system Components. The input-output transfer characteristics of many control-system components are often approximated by linear-system transfer operators of the following types ($p \equiv d/dt$):

$\dfrac{a_0}{\tau p + 1}$, $\dfrac{a_0}{\tau_1 p + 1}\dfrac{1}{\tau_2 p + 1}$ (e.g., THERMOCOUPLES AND OTHER TEMPERATURE SENSORS, ELECTROPNEUMATIC RELAYS, DIAPHRAGM MOTOR-VALVE ACTUATORS)

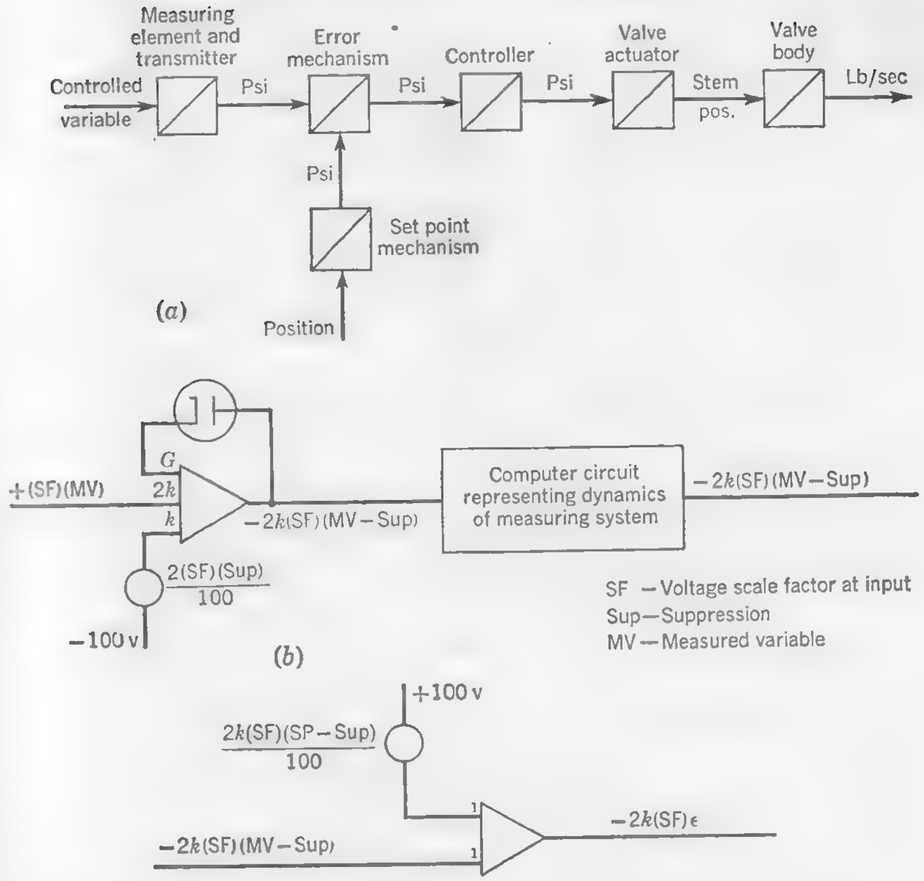


FIG. 5.7.3*a*. Block diagram of a pneumatic instrumentation-and-control system. For simplicity, the computer circuit will be scaled in terms of the measured variable (per cent scale readings), not in terms of actual pneumatic pressures.

FIG. 5.7.3*b*, *c*. Representation and scaling for a *suppressed-zero measuring instrument* which reads, say, temperature between 100 and 200°F (suppression = 100°); the limiter output is zero for simulated temperatures less than 100°. k is chosen so that the *instrument range* is just represented by −100 volt at the amplifier output (Fig. 5.7.3*b*). The suppression voltage is subtracted out in the summing amplifier representing the error mechanism (Fig. 5.7.3*c*).

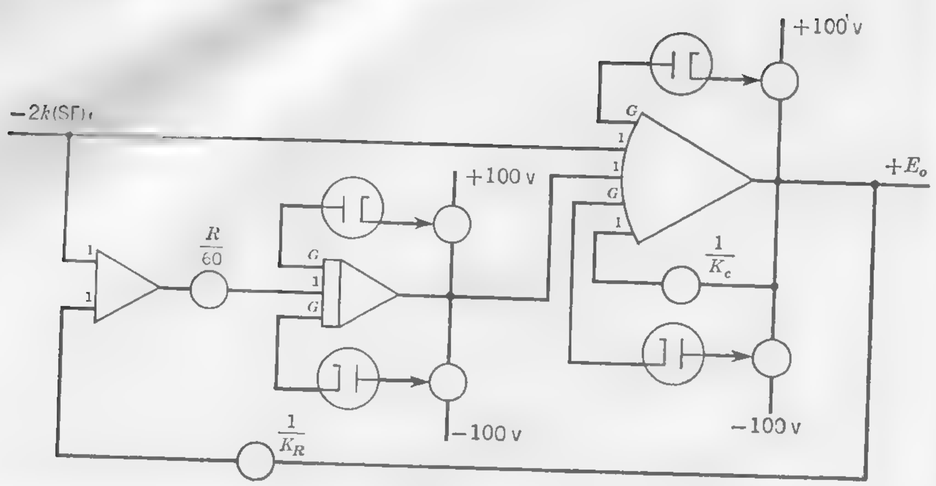
$\dfrac{a_0\omega_n^2}{p^2 + 2\zeta\omega_n p + \omega_n^2}$ (e.g., PRESSURE TRANSMITTERS, CONTROL VALVES, FLOW TRANSMITTERS, POTENTIOMETER RECORDERS)

$\dfrac{a_0}{\tau_1 p + 1}\dfrac{1}{\tau_2 p + 1}\dfrac{1}{\tau_3 p + 1}$ (FLOW TRANSMITTERS)

$\dfrac{a_0}{\tau p + 1}e^{-bp}$, $\dfrac{a_0}{\tau_1 p + 1}\dfrac{1}{\tau_2 p + 1}e^{-bp}$ (e.g., MAGNETIC OXYGEN ANALYZERS, THERMAL-CONDUCTIVITY GAS ANALYZERS, THICKNESS GAUGES)

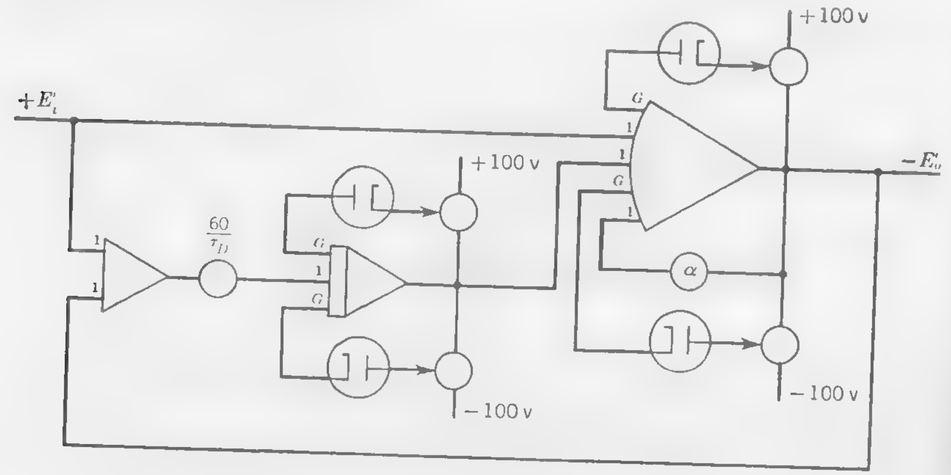
Specific transfer functions and parameter values must be obtained from manufacturers' specifications.[16] *Computer setups are easily obtained by reference to Tables* 2.2.1 *to* 2.2.4, with commercial transport-delay units representing each transport-delay operator e^{-ap} (Sec. 6.1).

Figures 5.7.2 and 5.7.4 show simple analog-computer representations of various industrial controllers (see also Fig. 5.7.3*d* and *e*). *Note carefully that most controller*

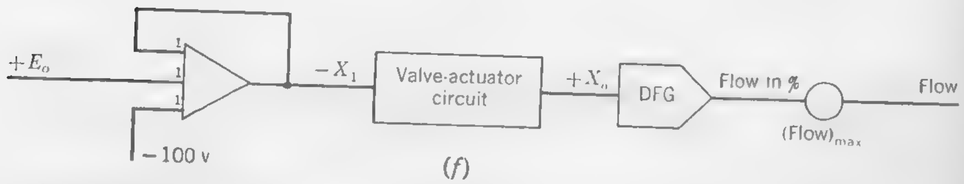


(*d*)

FIG. 5.7.3*d*. Representation and scaling for a theoretical *two-mode* (*proportional and reset*) *controller*. $R/60$ is the *reset rate* in repeats per second, and $100/K_c$ is the *proportional band* in per cent (Art. 5.7.3). Amplifier and integrator are diode-limited to ±100 volts, which represents the control-signal range.



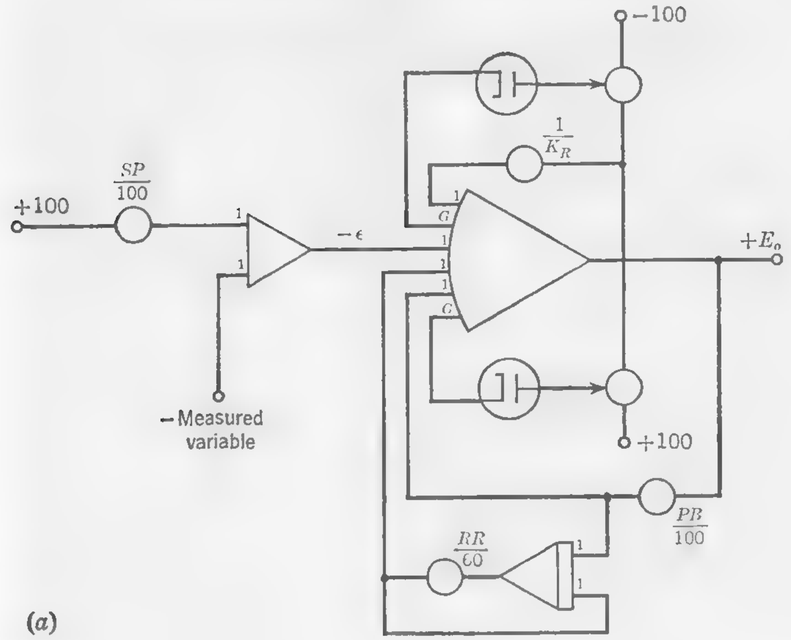
(*e*)



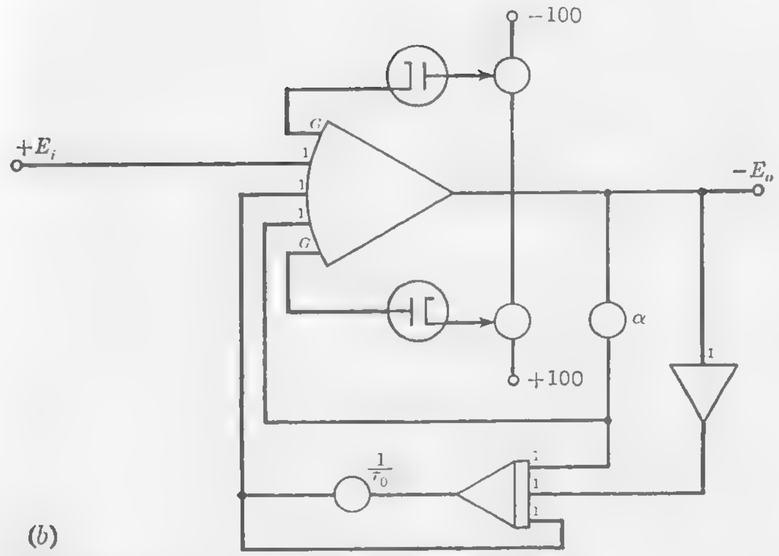
(*f*)

FIG. 5.7.3*e*. Representation of a *proportional-derivative controller* to be combined with Fig. 5.7.3*d*, if desired. Input and output scale factors are usually equal. $\tau_D/60$ is the rate time constant in minutes. Limiters are used in the manner of Fig. 5.7.3*d*. α is the rate stabilization gain.

FIG. 5.7.3*f*. Representation and scaling of a *valve actuator and control valve*. Since the valve is to operate between zero flow ($X_0 = 0$) and full flow ($X_0 = 100$ volts), a biasing amplifier converts the equalized error signal E_0 from the controller into an always-positive valve-command signal $X_1 = (E_0 + 100)/2$ and, incidentally, cancels the factor 2 from Fig. 5.7.3*d*. The pneumatic valve actuator is represented by a linear second-order circuit with $a_0 = 1$ (Art. 5.7.3) with or without limit stops (Sec. 5.3); X_0 represents the valve position in per cent of travel. The function generator and potentiometer are set to yield a voltage representing flow as function of valve position for each specific valve.



(*a*)



(*b*)

FIG. 5.7.4. Analog-computer representation of actual industrial-controller characteristics (neglecting high-frequency dynamic characteristics); (*a*) proportional plus reset (integral) controller; (*b*) proportional plus compensated derivative controller. This representation produces saturation characteristics which more closely correspond to that of an actual controller design. Scaling corresponds to that of Fig. 5.7.3*d* and *e*. Representative values of controller parameters are: $\frac{1}{10} < K_c < 100$; $0.3 < \tau_R < 600$ secs; $200 < K_R < 1500$; $0 < \tau_D < 1440$; $\frac{1}{25} < \alpha < \frac{1}{6}$; depending upon manufacturer.

specifications are given in terms of $S = 100/K_c$ (*proportional band in per cent*), $60/\tau_R$ (*reset rate in repeats per minute*), *and* $\tau_D/60$ (*rate time constant in minutes*).

5.7.4. Representation of a Typical Process-control System. Figure 5.7.3*a* shows a typical pneumatic valve-actuating control system. Its analog-computer representation (Fig. 5.7.3*b*, *c*, *d*) illustrates several setup and scaling techniques peculiar to proc-

Table 5.7.1

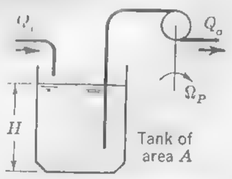
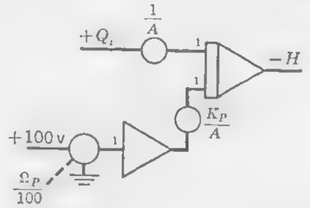
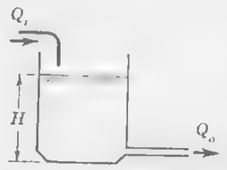
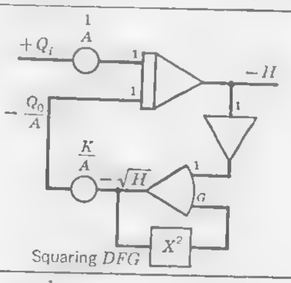
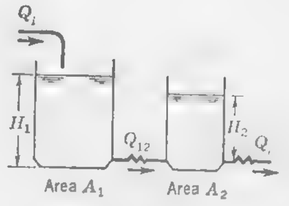
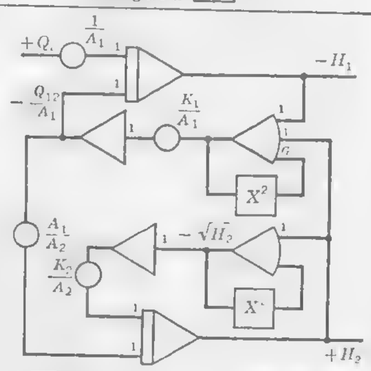
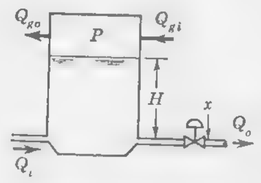
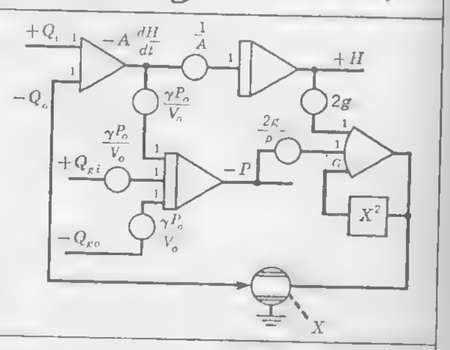
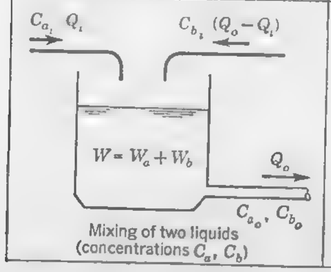
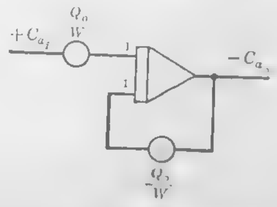
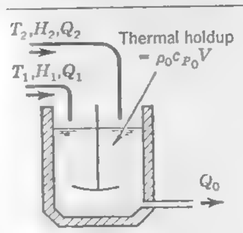
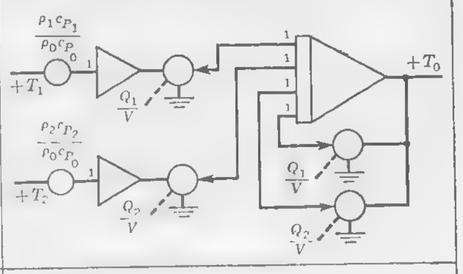
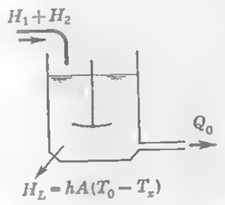
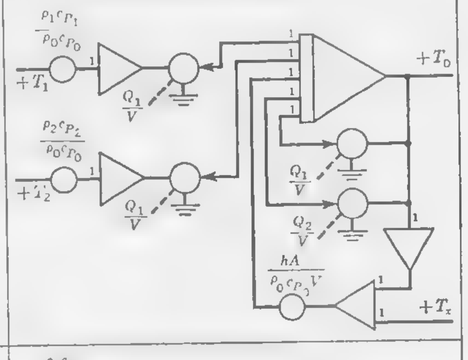
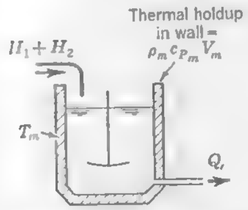
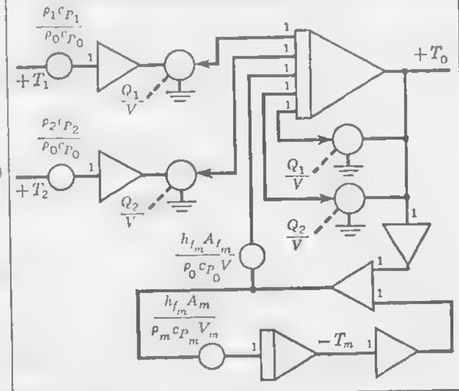
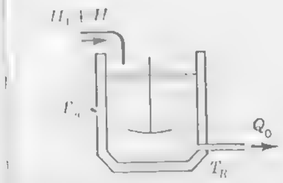
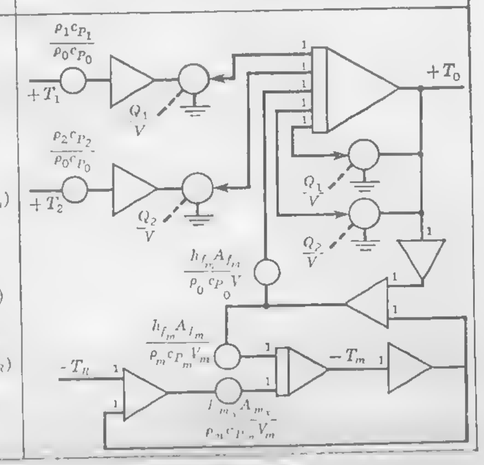
Process schematic	Differential equations	Computer circuit
 Liquid pumped from storage tank	$A\frac{dH(t)}{dt} = Q_i(t) - Q_o(t)$ $Q_o(t) = K_P\Omega_P(t)$ A = area of tank	
 Liquid flow through storage tank	$A\frac{dH(t)}{dt} = Q_i(t) - Q_o(t)$ $Q(t) = K\sqrt{H(t)}$	
 Liquid flow through two tanks	$A_1\frac{dH_1(t)}{dt} = Q_i(t) - Q_{12}(t)$ $A_2\frac{dH_2(t)}{dt} = Q_{12}(t) - Q_o(t)$ $Q_{12}(t) = K_1\sqrt{H_1(t) - H_2(t)}$ $Q_o(t) = K_2\sqrt{H_2(t)}$	
 Liquid flow with gas pressure	$A\frac{dH(t)}{dt} = Q_i(t) - Q(t)$ $Q_o(t) = f(x)\sqrt{2gH(t) - \frac{2P}{\rho}g}$ $C\frac{dP(t)}{dt} = Q_{gi} - Q_{go} - A\frac{dH}{dt}$ $C = \frac{V}{\gamma P_o}$	
 Mixing of two liquids (concentrations C_a, C_b)	$W\frac{dC_{a_o}}{dt} = Q_o(C_{a_i} - C_{a_o})$ (W = const)	

Table 5.7.2

Process schematic	Differential equations	Computer circuit
 Perfectly insulated and perfectly mixed vessel	$\rho_0 c_{P_0} V\frac{dT_0}{dt} = \rho_1 c_{P_1} T_1 Q_1 + \rho_2 c_{P_2} T_2 Q_2 - \rho_0 c_{P_0} T_0 Q_0$ $Q_0 = Q_1 + Q_2$	
 Perfectly mixed vessel with conducting zero- thermal-capacity wall	$\rho_0 c_{P_0} V\frac{dT_0}{dt} = \rho_1 c_{P_1} T_1 Q_1 + \rho_2 c_{P_2} T_2 Q_2 - \rho_0 c_{P_0} T_0 Q_0 - hA(T_0 - T_x)$	
 Perfectly mixed vessel insulated with finite-thermal-capacity wall	$\rho_0 c_{P_0} V\frac{dT_0}{dt} = \rho_1 c_{P_1} T_1 Q_1 + \rho_2 c_{P_2} T_2 Q_2 - \rho_0 c_{P_0} T_0 Q_0 - h_{f_m} A_{f_m}(T_0 - T_m)$ $\rho_m c_{P_m} V_m\frac{dT_m}{dt} = h_{f_m} A_{f_m}(T_0 - T_m)$	
 Perfectly mixed vessel with conduction losses and finite thermal-capacity wall	$\rho_0 c_{P_0} V\frac{dT_0}{dt} = \rho_1 c_{P_1} T_1 Q_1 + \rho_2 c_{P_2} T_2 Q_2 - \rho_0 c_{P_0} T_0 Q_0 - h_{f_m} A_{f_m}(T_0 - T_m)$ $\rho_m c_{P_m} V_m\frac{dT_m}{dt} = h_{f_m} A_{f_m}(T_0 - T_m) - h_{m_x} A_{m_x}(T_m - T_R)$	

ess-control applications. Real-time simulation is used in the example, but a time-scale change (Sec. 5.1) will be convenient in many applications.

PROCESS MODELS AND THEIR REPRESENTATION

5.7.5. Representation of Process Dynamics. Simplified *process models* involve:
1. *Linearization:* study of an approximately equivalent linear system, i.e., a combination of simple operators[56] such as $a_0/(\tau p + 1)$, $a_0\omega_n^2/(p^2 + 2\zeta\omega_n p + \omega_n^2)$, e^{-bp}.
2. *Replacement of distributed-parameter systems by lumped-parameter systems,* which can be described by ordinary differential equations. This is accomplished either "intuitively" by lumping of, say, zones in a reactor; or formally by difference techniques (Sec. 5.10).

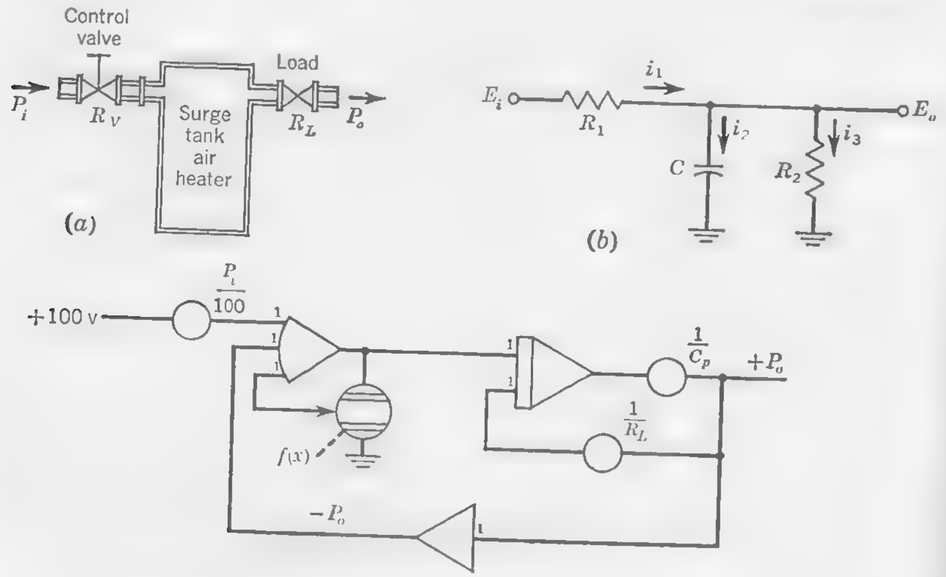


FIG. 5.7.5. Schematic diagram (*a*), network analogy (*b*), and analog-computer setup (*c*) for a process involving gas flow through a surge tank. Note Electronic Associates block-diagram notation used (Sec. 1.2).

Process variables in chemical, mixing, and transport processes (temperatures, pressures, inflow, outflow, concentrations of components, etc.) are related by differential equations and equations expressing basic physical relations in each vessel or zone:
1. *Material balance* for each component (conservation of mass)
2. *Heat balance* (conservation of energy)
3. *Thermodynamic state equations* (assuming equilibrium conditions)
4. *Reaction-rate equations,* which "cross-couple" material and heat balances
5. *Dynamic equations* governing fluid flow,[23,24] diffusion,[18,23,28] heat flow, etc.

Computer simulation may lead to parameter optimization for stable operation and optimum yield, and to development of safe and economical startup and shutdown procedures.

5.7.6. Some Process Models and Computer Representations. Tables 5.7.1 and 5.7.2 illustrate representative process relations involving *fluid flow through tanks* and *convection heating.* The symbols employed, in addition to miscellaneous constant parameters, are

A area, sq ft
C capacity of a vessel, lb per psi, or fractional concentration of a component
H height of fluid column ft, or heat flow, Btu per sec
h heat-loss coefficient (film coefficient)
P pressure, psi
q pump flow rate
Q flow rate, cu ft per sec or lb per sec
R effective resistance of a restriction, psi per lb per sec
T temperature, °F
V volume, cu ft
W material hold-up in a vessel, lb
x valve displacement, per cent
ρ density, lb per cu ft
c_p specific heat
t time, sec

with suitable subscripts added to distinguish between two or more quantities of the same type.

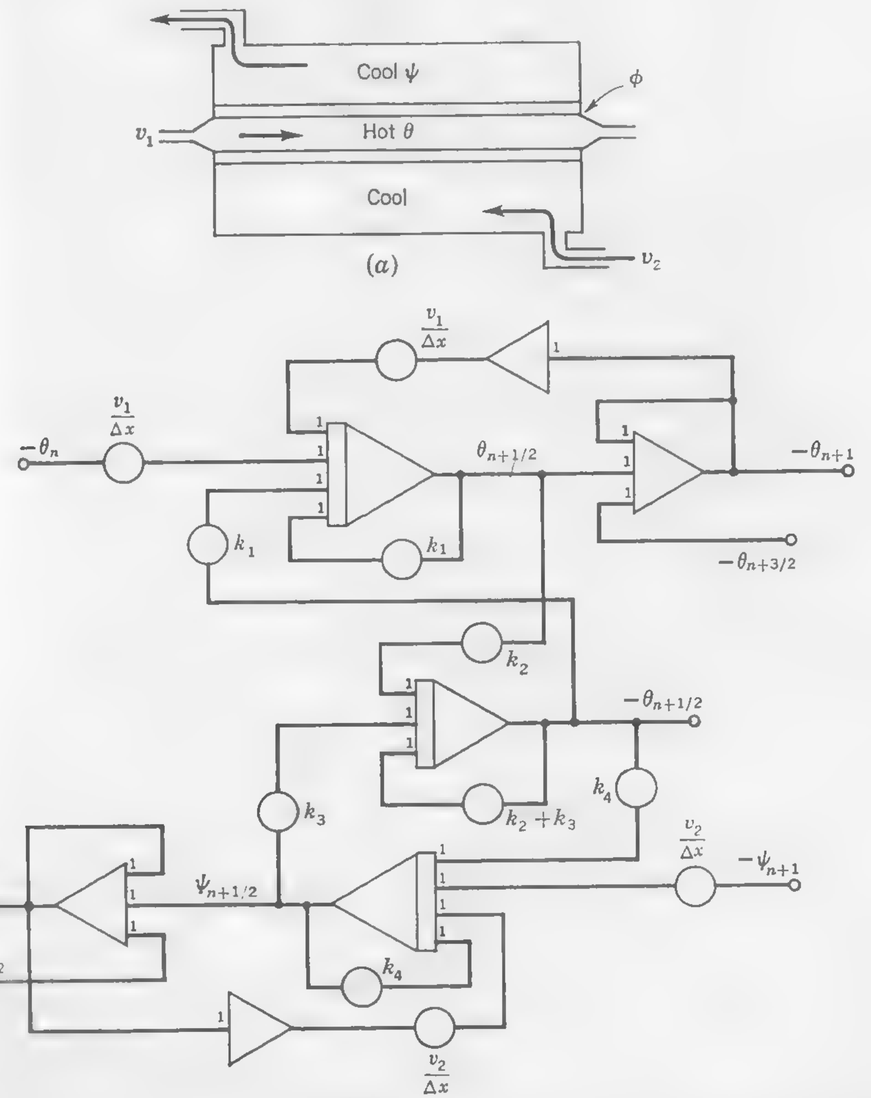


FIG. 5.7.6. Schematic diagram of a counterflow concentric-tube heat exchanger (*a*), and computer representation for one cell (*b*).

Note that each process is represented by a computer block whose input and output variables may be connected by control-system blocks to complete a process-control loop.

Some process analysts derive fluid-flow relations from *direct analogy*[22] with corresponding electrical networks. Figure 5.7.5 shows an equivalent network and an analog-computer setup representing *gas flow through a surge tank.*

5.7.7. A Process Model Involving Partial Differential Equations: Representation of a Heat Exchanger (see also Sec. 5.10).[23,26,27] Figure 5.7.6*a* shows a counterflow concentric-tube *heat exchanger.* The heat flow is described by the partial differential equations

$$\begin{aligned}
\frac{\partial\theta}{\partial t} + v_1\frac{\partial\theta}{\partial x} &= \frac{\pi D h_1}{\rho_1 c_1 A_1}(\phi - \theta)\\
\frac{\partial\phi}{\partial t} &= \frac{h_1}{\rho_w c_w x_w}(\theta - \phi) + \frac{h_2}{\rho_w c_w x_w}(\psi - \phi)\\
\frac{\partial\psi}{\partial t} - v_2\frac{\partial\psi}{\partial x} &= \frac{\pi D h_2}{\rho_2 c_2 A_2}(\phi - \psi)
\end{aligned} \tag{5.7.1}$$

where θ = temperature of hot fluid
ϕ = temperature of wall between fluids
ψ = temperature of coolant
v_1 = rate of flow of hot fluid
v_2 = rate of flow of coolant
D = diameter of tube
h_1 = film coefficient, hot liquid to wall
ρ_1 = density of hot liquid
c_1 = specific heat of hot l quid
A_1 = flow cross section of inner tube
ρ_w = density of wall
c_w = specific heat of tube wall
x_w = thickness of tube wall
h_2 = film coefficient, wa l to coolant flow
ρ_2 = dens ty of coolant
c_2 = specific heat of coolant
A_2 = flow cross section of coolant
x = dimension along length of exchanger

The analog-computer setup of Fig. 5.7.6*b* solves these partial differential equations in terms of the finite-difference approximation

$$\begin{aligned}
\frac{d\theta_{n+\frac{1}{2}}}{dt} &= \frac{v_1}{\Delta x}(\theta_n - \theta_{n+1}) + k_1(\phi_{n+\frac{1}{2}} - \theta_{n+\frac{1}{2}})\\
\frac{d\theta_{n+\frac{1}{2}}}{dt} &= k_2(\theta_{n+\frac{1}{2}} - \phi_{n+\frac{1}{2}}) + k_3(\psi_{n+\frac{1}{2}} - \phi_{n+\frac{1}{2}})\\
\frac{d\psi_{n+\frac{1}{2}}}{dt} &= \frac{v_2}{\Delta x}(\psi_{n+1} - \psi_n) + k_4(\phi_{n+\frac{1}{2}} - \psi_{n+\frac{1}{2}})
\end{aligned} \tag{5.7.2}$$

where

$$\begin{aligned}
k_1 &= \frac{\pi D h_1}{\rho_1 c_1 A_1} \qquad k_2 = \frac{h_1}{\rho_w c_w x_w}\\
k_3 &= \frac{h_2}{\rho_w c_w x_w} \qquad k_4 = \frac{\pi D h_2}{\rho_2 c_2 A_2}
\end{aligned} \tag{5.7.3}$$

Suitable boundary conditions must be given. For a majority of applications five cells give sufficient (about 4 per cent) accuracy. Similar techniques also permit analog-computer solutions of *nonlinear* partial differential equations, e.g., those describing fluid flow through long pipes.[57]

5.7.8. Simulation of a Distillation Column.[29 to 34] Figure 5.7.7*a* shows a simplified schematic diagram of a *binary distillation column* designed to separate two components of different volatility. The output concentrations x_n, y_m of the first component in the

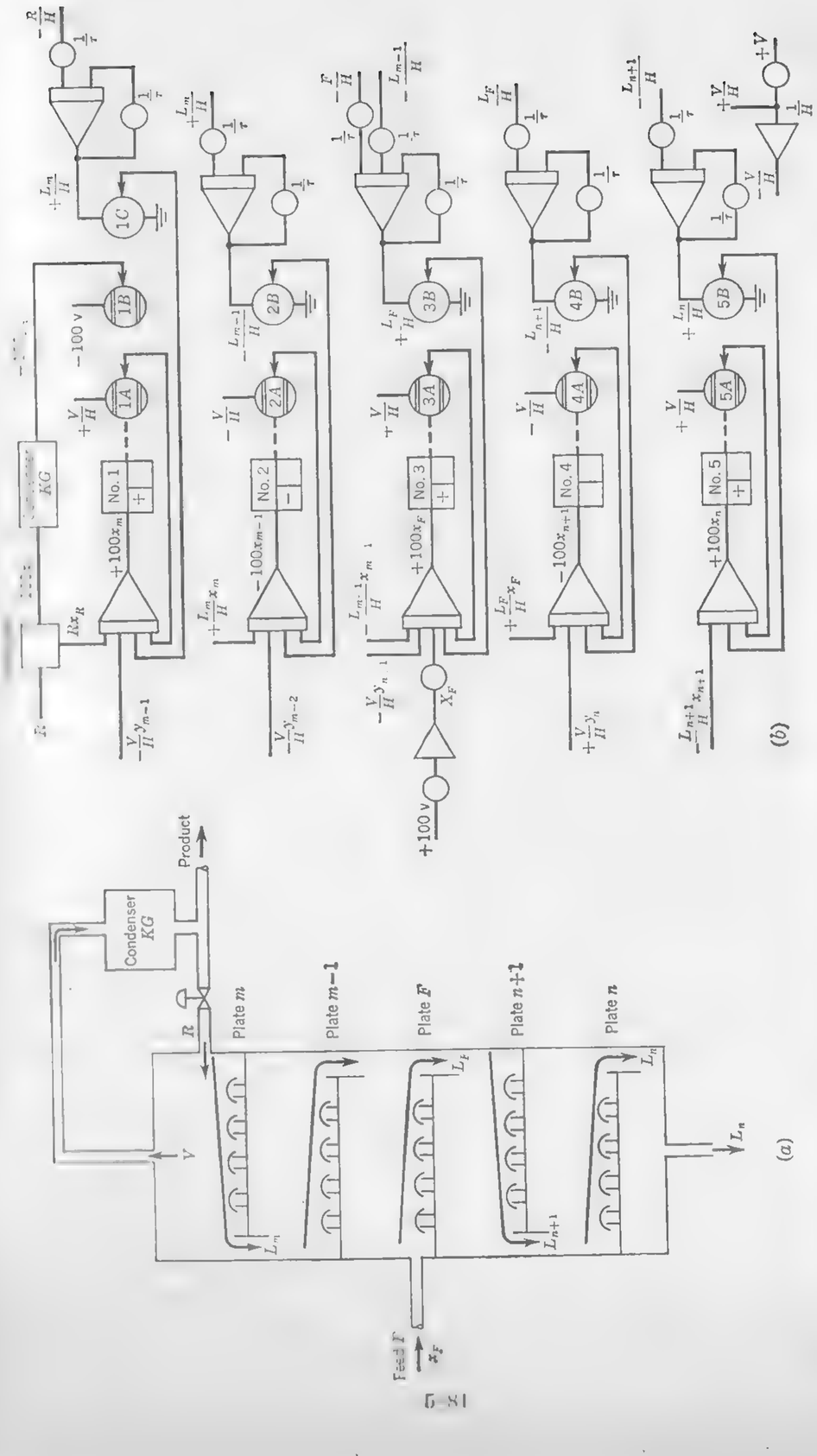


FIG. 5.7.7. Schematic diagram of an idealized five-plate binary distillation column (*a*), and computer representation (*b*). See Sec. 1.2 for the Electronic Associates servo notation used here; barred circles stand for tapped function potentiometers (Sec. 3.1).

liquid and vapor outputs are determined by the feed and reflux flows and by various design parameters.

The problem involves material balance between liquid and gas phases at each plate:

$$H\frac{dx_m}{dt} = Vy_{m-1} + Rx_R - Vy_m - L_m x_m \qquad \text{(PLATE } m\text{, TOP)}$$

$$H\frac{dx_{m-1}}{dt} = Vy_{m-2} + L_m x_m - L_{m-1}x_{m-1} - Vy_{m-1} \qquad \text{(PLATE } m-1\text{)}$$

$$H\frac{dx_F}{dt} = Vy_{F-1} + L_{F+1}x_{F+1} - L_F x_F - Vy_F + Fx_{\text{feed}} \qquad \text{(FEED PLATE } F\text{)}$$

$$H\frac{dx_{n+1}}{dt} = Vy_n + L_{n+2}x_{n+2} - L_{n+1}x_{n+1} - Vy_{n+1} \qquad \text{(PLATE } n+1\text{)}$$

$$H\frac{dx_n}{dt} = L_{n+1}x_{n+1} - L_n x_n - V_n y_n \qquad \text{(PLATE } n\text{, BOTTOM)}$$

Liquid flow between plates is described by

$$\frac{L_n}{L_{n+1}} = \frac{1}{\tau p + 1}$$

(see also Table 5.7.1); the vapor flow V is assumed to be constant. Vapor equilibrium concentrations and reflux concentration are described by

$$y = \frac{\alpha x}{1 + (\alpha - 1)x} \quad \text{(EACH PLATE)} \qquad x_R = KGy_m \quad \text{(TOTAL CONDENSER)}$$

The following symbols are employed:

V = vapor flow, moles per sec (assumed constant through column)
R = reflux flow, moles per sec
L = liquid flow, moles per sec
x = liquid composition in mole fraction
y = vapor composition in mole fraction
F = feed-flow rate, moles per sec
x_F = feed composition in mole fraction
τ = liquid-plate hold-up time constant
α = relative volatility
KG = overhead-condenser transfer function

Figure 5.7.7*b* shows a computer setup for these equations; the setup may be expanded to include additional plates. Representation of each plate in the column would require a very large amount of computing equipment, particularly for multicomponent columns (300 to 500 amplifiers and associated nonlinear equipment). As a result, one attempts to replace groups of plates by equivalent theoretical plates for purposes of analysis; but the author knows of no published information regarding the success or failure of this approach.

PROCESSES INVOLVING CHEMICAL REACTIONS

5.7.9. Introduction. Practical questions about a chemical reaction involve two fundamental factors: *equilibrium yield* and *rate of reaction*. *Reaction-rate equations* express concentration changes of each component in terms of state variables such as temperature, pressure, concentrations of components, and concentration and age of catalysts. The differential equations expressing heat and material balances are coupled by reaction-rate terms. The formulation of the heat- and material-balance equations will follow the procedures already discussed. These equations must be modified to the extent of including terms which specify the amount of heat energy added or absorbed by the reaction (*heat of reaction*); and, of course, the material balance must

account for the material lost or gained through the reaction. For instance, a general[35] material-balance equation for processes involving chemical reactions is

$$\begin{pmatrix}\text{Reactant}\\ \text{added in}\\ \text{feed stream}\end{pmatrix} - \begin{pmatrix}\text{reactant}\\ \text{lost in}\\ \text{exit stream}\end{pmatrix} - \begin{pmatrix}\text{reactant}\\ \text{converted}\\ \text{in reactor}\end{pmatrix} = \begin{pmatrix}\text{rate of change}\\ \text{of reactant}\\ \text{in reactor}\end{pmatrix}$$

Table 5.7.3 illustrates a classification of the equipment (reactors) used to contain chemical-reaction processes.

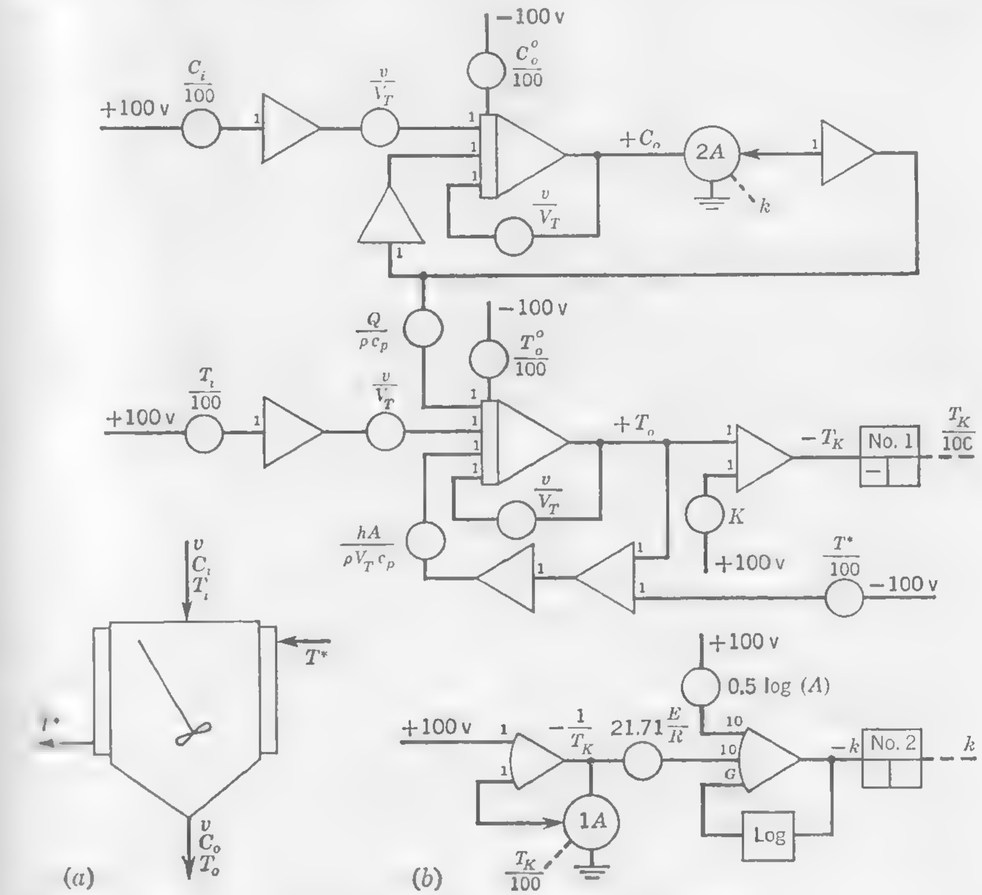


FIG. 5.7.8. Stirred-tank reactor (*a*) and analog-computer representation of a chemical process (*b*). The inverse-function-generator circuit at the bottom of Fig. 5.7.8*b* generates the specific reaction rate $k = A \exp(-E/RT_K)$ as a function of the absolute process temperature $T_K = T + 273$ (*Arrhenius equation*).

5.7.10. Analog-computer Representation of a Chemical Process. Let C_i and C_o be the respective concentrations (moles per cu ft) of two components A and B. For a simple chemical reaction $A \rightarrow B$, the reaction-rate equation is

$$\frac{dC_o}{dt} = -\frac{dC_i}{dt} = r = kC_i \qquad (5.7.4)$$

The *specific reaction rate* k is usually approximated by

$$k = ae^{-E/RT_K} \qquad \text{(ARRHENIUS EQUATION)} \quad (5.7.5)$$

where a, E, and R are constants, and T_K is the absolute temperature; k may also depend on pressure, presence of catalysts, etc.

Table 5.7.3. Chemical-reactor Classification*

	Batch	Homocontinuous	Heterocontinuous
Example	Stirred tank	CFSTR, fluid bed	Tubular: fixed and moving bed
Phases involved	Usually liquid or gas	Usually liquid or gas with/without solid	Fluid and solid
Independent variables	Time	Time	Time and space
Uses and products	Kinetic studies, acrylic monomers, analine, emulsion and solution, polymerization	Kinetic studies, gasoline, polymers, activated or regenerated catalyst	Kinetic studies; partial oxidation, SO_2, gasoline treating, bulk polymerization, flames, metalscasting, ion-exchange beds, blast furnaces

* G. Marr, Analog Computer Solution of Chemical Reactor Problems, *Application Bull.* 4, Electronic Associates, Inc., Long Branch, N.J.

In the stirred-reactor process of Fig. 5.7.8*a*, the chemical reaction is again $A \rightarrow B$. The reaction-rate equation must be entered into the material-balance equation, and the heat of reaction affects the heat balance:

1. *Material Balance**

Material in	vC_i
Material out	vC_o
Material reacted	kV_TC_o
Material accumulated	$V_T \dfrac{dC_o}{dt}$

Summing:

$$\frac{dC_o}{dt} = \frac{v}{V_T} C_i - \frac{v}{V_T} C_o - kC_o$$

2. *Heat Balance*

Heat in	$v\rho c_p T_i$
Heat out	$v\rho c_p T_o$
Heat transferred	$-hA(T_o - T^*)$
Heat due to reaction	$-QV_TkC_o$
Heat accumulated	$\rho V_T c_p \dfrac{dT_o}{dt}$

Summing:

$$\frac{dT_o}{dt} = \frac{v}{V_T} T_i + \frac{hA}{\rho V_T c_p} (T^* - T_o) - \frac{QkC_o}{\rho c_p} - \frac{v}{V_T} T_o$$

The computer circuit for solving the three equations is shown in Fig. 5.7.8*b*; the following symbols are employed:

v = input and output flow rate (assumed constant)
C_i = concentration of input stream
C_o = concentration of output stream
k = specific reaction rate
V_T = total volume of reaction (assumed constant)
ρ = density of reactant and product (assumed constant)
c_p = specific heat of reactant and product (assumed constant)

* The hold-up in the tank is assumed to be constant, as is the density.

T_i = temperature of input stream
T_o = temperature of output stream
h = coefficient of heat transfer
A = heat-transfer area
Q = heat of reaction
T^* = temperature of jacket

5.7.11. More General Reaction-rate Equations. For a more general, *reversible* reaction $A + B \rightleftarrows C + D$ in a single phase at constant volume, denote the component concentrations by C_A, C_B, C_C, C_D (moles per cu ft). For a so-called *second-order* reaction the net reaction rate for component A (net rate of increase of C_A) is the difference between the rates of forward and reverse second-order reactions, i.e.,

$$r = k_1C_AC_B - k_2C_CC_D \tag{5.7.6}$$

where k_1 and k_2 are specific reaction rates for the forward and reverse reactions. Equation (6) is used like equation (3) but will, in general, require more nonlinear computing equipment.

References

1. Fuchs, A. M.: A Bibliography of Frequency Response Method as Applied to Automatic Feedback Control Systems, ASME Paper 53-A-13, presented at ASME-IRD Frequency Response Symposium, New York, December, 1953.
2. Truxal, J. G.: *Automatic Feedback Control System Synthesis*, McGraw-Hill Book Company, Inc., New York, 1955.
3. Brown, G. S., and D. P. Campbell: *Principles of Servomechanisms*, John Wiley & Sons, Inc., New York, 1948.
4. Chestnut, H., and R. W. Mayer: *Servomechanism and Regulating System Design*, vol. 1, John Wiley & Sons, Inc., New York, 1951.
5. Thaler, G. J., and R. G. Brown: *Servomechanism Analysis*, McGraw-Hill Book Company, Inc., New York, 1953.
6. Rutherford, C. I.: The Practical Application of Frequency Response Analysis to Automatic Process Control, *Proc. Inst. Mech. Engrs.*, **162**:334–343 (1950).
7. Tustin, A.: *Automatic and Manual Control*, Academic Press, Inc., New York, 1952.
8. Farrington, G. H.: *Fundamentals of Automatic Control*, John Wiley & Sons, Inc., New York, 1951.
9. Ceaglske, N. H.: *Automatic Process Control for Chemical Engineers*, John Wiley & Sons, Inc., New York, 1956.
10. Kochenburger, R. J.: A Frequency Response Method for Analyzing and Synthesizing Contactor Servomechanisms, *Trans. AIEE*, **69**:270–284 (1950).
11. MacColl, L. A.: *Fundamentals Theory of Servomechanisms*, D. Van Nostrand Company, Inc., Princeton, N.J., 1945.
12. Ahrendt, W. R., and J. F. Taplin: *Automatic Feedback Control*, McGraw-Hill Book Company, Inc., New York, 1951.
13. Oldenbourg, R. C., and H. Sartorius: *The Design of Automatic Controls*, translated by H. L. Mason, American Society of Mechanical Engineers, New York, 1948.
14. Ziegler, J. G., and N. B. Nichols: Optimum Settings for Automatic Controllers, *Trans. ASME*, **65**:5, 433 (1943).
15. Pessen, D. W.: Optimum Three-mode Controller Settings for Automatic Start-up, ASME Paper 52-A-58.
16. Eckman, D. P.: *Automatic Process Control*, John Wiley & Sons, Inc., New York, 1958.
17. Karplus, W. J.: *Analog Simulation*, McGraw-Hill Book Company, Inc., New York, 1959.
18. Hougen, O. A., and K. M. Watson: *Chemical Process Principles*—Part III, *Kinetic and Catalysis*, John Wiley & Sons, Inc., New York, 1947.
19. Goodman, T. P., and J. B. Reswick: Determination of System Characteristics from Normal Operating Records, ASME Paper 55-IRD-1.
20. Cowley, P. E. A.: The Application of an Analog Computer to the Measurement of Process Dynamics, ASME Paper 56-IRD-20.
21. Baker, H. D., and V. Paschkis: A Method of Determining Unsteady-state Heat Transfer by Means of an Electrical Analogy, *Trans. ASME*, 1942.
22. Larrowe, V. L.: Direct Simulation Bypasses Mathematics, Simplifies Analysis, *Control Eng.*, November, 1954.
23. Campbell, D. P.: *Process Dynamics*, John Wiley & Sons, Inc., New York, 1958.

24. Badger, W. L., and W. L. McCabe: *Elements of Chemical Engineering*, McGraw-Hill Book Company, Inc., New York, 1936.
25. Brown, A. I., and S. M. Marco: *Introduction to Heat Transfer*, 3d ed., McGraw-Hill Book Company, Inc., New York, 1958.
26. Jacob, M.: *Heat Transfer*, vol. 1, John Wiley & Sons, Inc., New York, 1949.
27. Takahashi, Y.: Transfer Function Analysis of Heat Exchangers, in *Automatic and Manual Control*, edited by A. Tustin, Academic Press, Inc., New York, 1952.
28. Jeans, J. H.: *Dynamical Theory of Gases*, 3d ed., Cambridge University Press, New York, 1921.
29. Marshall, W. R., and R. L. Pigford: *The Application of Differential Equations to Chemical Engineering Problems*, University of Delaware, Newark, Del., 1947.
30. Armstrong, W. D., and W. L. Wilkinson: An Investigation of the Transient Response of a Distillation Column, in *Plant and Process Dynamic Characteristics*, Academic Press, Inc., New York, 1957.
31. Aikman, A. R.: Frequency Response Analysis of a Fractionating Column, *ISA Journal*, October, 1946, p. 412.
32. Voetter, H.: Response of Concentration in a Distillation Column to Disturbances in the Feed Composition, in *Plant and Process Dynamic Characteristics*, Academic Press, Inc., New York, 1957.
33. Teager, H. M.: An Analysis of Transients in Mass-transfer Operations, MIT Sc.D. Thesis, 1955.
34. Williams, T. J.: An Investigation of the Automatic Control of a Distillation Column, Pennsylvania State University Sc.D. Thesis, 1956.
35. Smith, J. M.: *Chemical Engineering Kinetics*, McGraw-Hill Book Company, Inc., New York, 1956.
36. Denbigh, K. G.: *Trans. Faraday Soc.*, **40**:352 (1944).
37. Bubb, F. W., and W. L. Morris: How Analogical Computing Devices Can Serve Process Industries, *Chem. Eng.*, **57**:142 (July, 1950).
38. Morris, W. L.: Analogical Computing Devices in the Petroleum Industry, *Ind. Eng. Chem.*, **43**:2478 (November, 1951).
39. Isaac, A. H.: An Approach to Automatic Process Control, *Trans. Chem. Eng.*, **29**(3), (1951).
40. Simulation of a Process Controller, Goodyear Aircraft Corp., Akron, Ohio, *Rept.* GER-4689, March, 1952.
41. Matthews, H., and R. J. Medkeff: Solving Process Control Problems by Analog Computer, *Instr. and Automation*, **27**:1624 (1954).
42. Murphy, G.: Test Models, *Chem. Eng.*, **61**:183 (August, 1954).
43. Goldstein, W. A.: Automatic Control of Batch Systems, *Trans. Inst. Chem. Engrs.*, **33**(3), (1955).
44. Johnson, E. F.: Analogs as Aids in Process Studies Introduction, *Ind. Eng. Chem.*, **27**(3): 396 (1955).
45. Lefkowitz, I., and T. J. Walsh: Optimizing Control of a Distillation Column, *ISA Journal*, **2**(1), (1955).
46. Smith, B. O.: Automatic Control in the Steel Industry, *Trans. Ind. Chem. Eng.*, **33**(3): 195 (1955).
47. Hussey, J. L.: Process Control and the Analog Computer, *Instr. and Automation*, **29**(10): 2000 (1956).
48. Amundson, N. R., H. S. Isbin, and E. H. Wissler: Oscillatory Behavior of a Two-phase Natural-circulation Loop, *A.I.Ch.E. Journal*, **2**(2): 157 (1956).
49. Woods, F. A.: Simulation of Process Control with an Analog Computer, *Ind. Eng. Chem.*, **50**(11): 1627 (1958).
50. Field, W. B.: Design of a pH Control System by Analog Simulation, *ISA Journal*, January, 1959.
51. Franks, R. W., J. F. Pink, and C. W. Worley: Process Control Problems Yield to Analog Computer, *Control Eng.*, **4**(6): 97 (1957).
52. Bilous, O., H. D. Block, and E. L. Piret: Control of Continuous Flow Chemical Reactor, *A.I.Ch.E. Journal*, **3**(2): 248 (1957).
53. Batke, T. L., R. W. Franks, and E. W. James: Analog Computer Simulation of a Chemical Reactor, ISA Paper 56-7-2.
54. Williams, T. J.: Automation Today, part III, *Petrol. Refiner*, **36**(8): 165 (1957).
55. Considine, D. M.: *Process Instruments and Control Handbook*, McGraw-Hill Book Company, Inc., New York, 1957.
56. Truxal, J. G.: *Control Engineers' Handbook*, chap. 19, McGraw-Hill Book Company, Inc., New York, 1958.
57. Vichnevetsky, R.: *European Computation Center Rept.* 20, Electronic Associates, Inc., Long Branch, N.J.

PART 8 : NUCLEAR-REACTOR SIMULATION

By E. Morrison

5.8.1. Introduction. Electronic analog computers can simulate the time behavior of a nuclear reactor or power plant on a suitable time scale. The relevant equations describing reactor neutron kinetics, poison build-up and decay, and basic reactor thermodynamics are outlined in the following sections. For the nonlinear parameter considerations necessary to describe heat-transfer and hydrodynamic properties in the various types of power plants more accurately the reader is referred to the list of references at the end of this section.

Table 5.8.1. Glossary of Symbols

ϕ	neutron flux	ω	coolant mass-flow rate
Sk	excess core reactivity	T_o	core-coolant outlet temperature
β	total delayed neutron fraction	T_i	core-coolant inlet temperature
l^x	prompt neutron lifetime	τ_c	coolant transit time in core
C_i	concentration of the i^{th} delay group	T_{ix}, T_{ox}	heat-exchanger inlet and outlet temperatures
λ_i	decay constant of the i^{th} delay percursor	$\bar{T}_m$	average heat-exchanger piping temperature
β_i	fraction of flux delayed by i^{th} group	T_s	average secondary coolant temperature
s	neutron source term	$(mc)_s$	secondary coolant thermal capacity
P	core generated power	Km_s	heat-transfer coefficient; piping walls to secondary coolant
$(mc)_f$	fuel thermal capacity	I	iodine 131 concentration
$\bar{T}_f$	average fuel temperature	Xe	xenon 135 concentration
K_f	fuel-element-surface heat-transfer coefficient	λ_I	iodine decay constant
$\bar{T}_c$	average core-coolant temperature	λ_{Xe}	xenon decay constant
$(mc)_c$	core-coolant thermal capacity		

5.8.2. Representation of Nuclear Kinetics. The basic equations describing the time variations of average neutron flux in a reactor are

$$\frac{d\phi}{dt} = \frac{\delta K - \beta}{l_x}\phi + \sum_{i=1}^{6} \lambda_i C_i + s$$
$$\frac{dC_i}{dt} = \frac{\beta_i \phi}{l^x} - \lambda_i C_i \qquad (5.8.1)$$

The parameters in the above equations are identified in Table 5.8.1. Values of delayed-neutron fractions B_i and corresponding decay constants λ_i for a U 235 core are listed in Table 5.8.2.

Table 5.8.2. Delayed-neutron Fractions B_i and Decay Constants λ_i for a U 235 Core

Group	B_i	λ_i, sec^{-1}
1	8.4×10^{-4}	1.61
2	2.4×15^{-3}	0.456
3	2.1×15^{-3}	0.151
4	1.7×10^{-3}	3.15×10^{-2}
5	2.6×15^{-4}	1.24×10^{-2}

If only subcritical operation of the pile is to be demonstrated, the circuit of Fig. 5.8.1 will adequately represent the time behavior characteristics of the average flux. This

is only the average flux; spatial variations are not considered. The six RC networks represent the contribution of the six delayed neutron groups to total flux. The time constants represent the λ_i quantities in Table 5.8.2, and the individual currents represent the percentage contribution of each delayed percursor. The capacitor C_1 represents l^x in the previous kinetic equations. P_1 simulates the control rod used for reactivity variation.

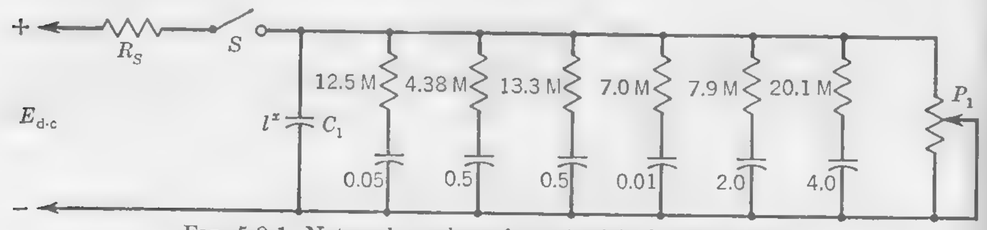


FIG. 5.8.1. Network analogy for subcritical neutron flux.

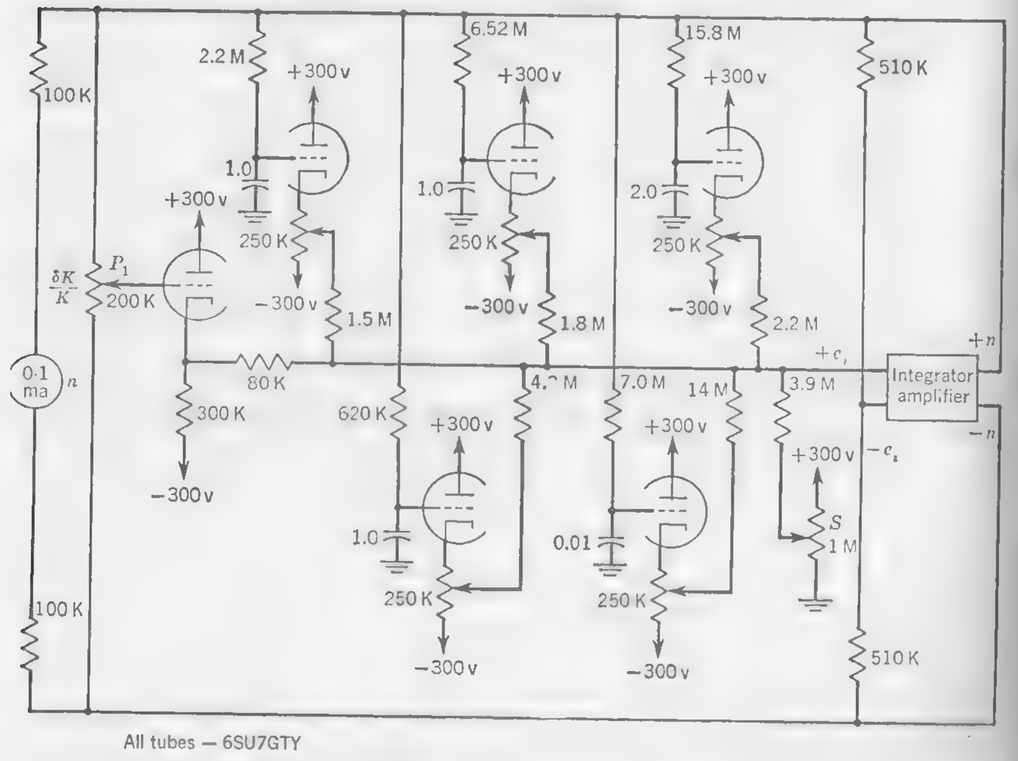


FIG. 5.8.2. Bell-Strauss pile simulator (*from Ref. 7*).

Spatial flux distribution within a core volume can be represented by using a number of these passive-network simulators driven from one or more voltage sources to correspond to the number of neutron sources in the core. These networks are coupled so as to represent flux leakage between volume sections of the core. The potentiometers used in each circuit can be calibrated to simulate the value of a single central control rod or the induced reactivity variations in the separate core sections.

When the reactor is to be simulated in the dynamic power range, operational amplifiers offer many advantages for representation of neutron kinetics. The Bell-Strauss simulator shown in Fig. 5.8.2 was one of the first active-kinetics simulators.[7] The operational element is a double-ended integrating amplifier. Reactivity is varied by potentiometer P. The six input RC networks represent the percentages and decay constants of the delayed neutron groups.

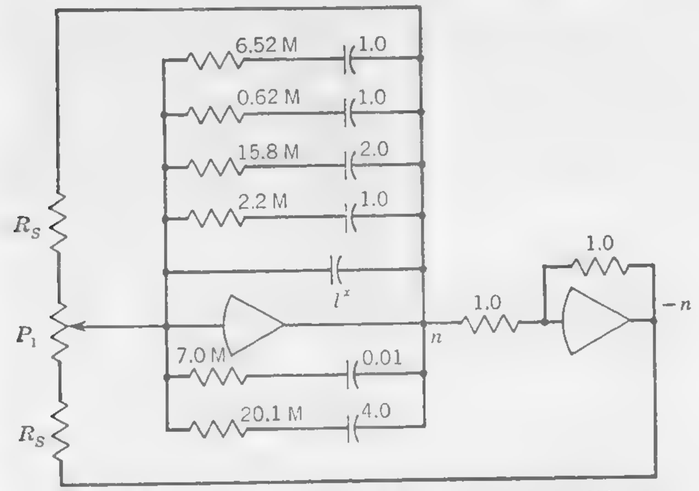


FIG. 5.8.3. Pile simulator using operational amplifiers.

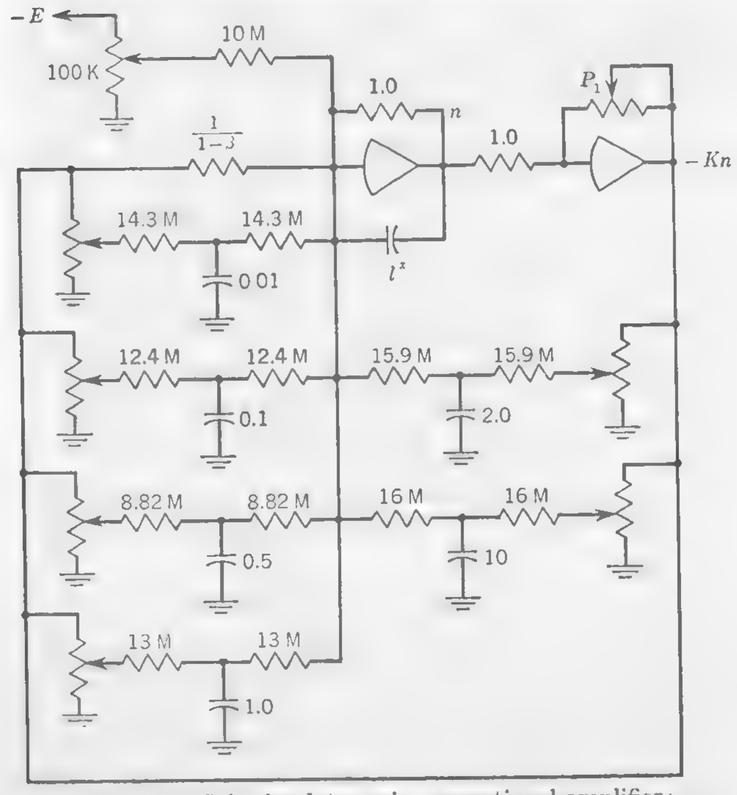


FIG. 5.8.4. Pile simulator using operational amplifiers.

The amplifier is a high-gain differential unit with two cathode followers in the output circuit. An 0.1-μf integrating capacitor is connected between the $-n$ output and positive input terminals. The source potentiometer is used to represent core neutron sources and also to compensate for any residual voltage differences between the various input cathode followers.

Just as with any high-gain circuit care must be exercised in construction and shielding to prevent parasitic oscillations, especially if one requires remote operation of the reactivity potentiometer.

Figures 5.8.3 and 5.8.4 show similar techniques for kinetics simulation using conventional single-ended operational amplifiers. The first circuit is similar to that of Fig. 5.8.2. The networks representing the delayed-neutron fractions and decay constants are here in the amplifier feedback circuit. Reactivity is again varied by the potentiometer P_1. The simulator of Fig. 5.8.4 is similar to that just described, but the delayed-neutron networks are used in the input circuit of amplifier 1. The reactivity-control function is obtained simply through variation of loop gain by P_1.

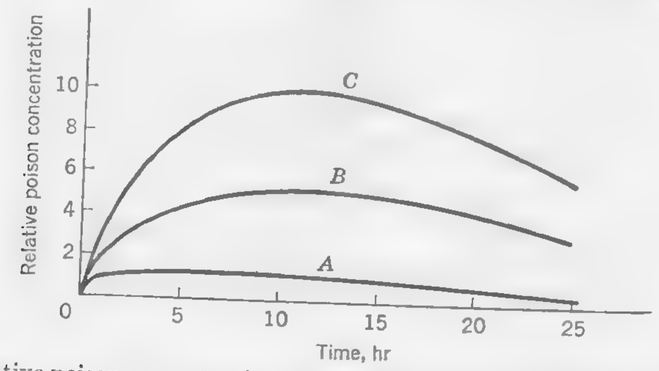


FIG. 5.8.5. Relative poison concentration vs. time for $\phi = 10^{13}$ N per cm² per sec (curve A), $\phi = 10^{14}$ N per cm² per sec (curve B), and $\phi = 2 \times 10^{14}$ N per cm² per sec (curve C).

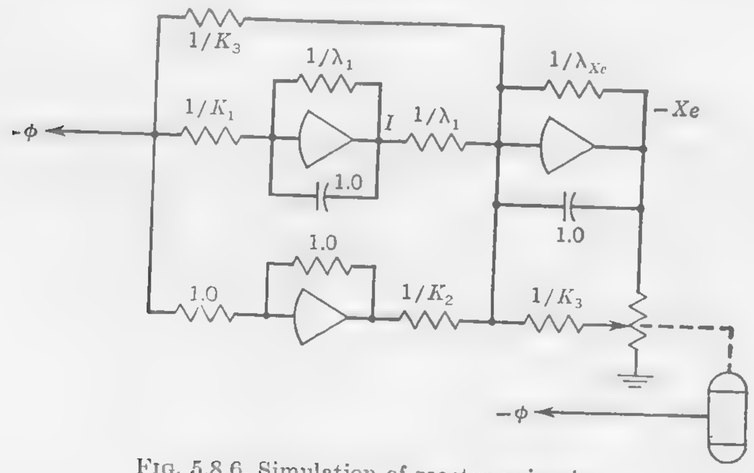


FIG. 5.8.6. Simulation of reactor poisoning.

In all the active simulators described, the reactivity-control potentiometer P_1 can be servo-driven with an adder summing all reactivity inputs (absorber rods, temperature coefficients, poison effects, void fractions) so that the potentiometer position could represent total average multiplication in a complete or section core volume.

5.8.3. Representation of Poison Generation. Reactor operation generates "poisons" which affect reactivity, primarily iodine 131 and xenon 135. The rate of generation for these fission products is given by

$$\begin{aligned} \frac{dI}{dt} &= K_1\phi - \lambda_1 I \\ \frac{dXe}{dt} &= K_2\phi + \lambda_1 I - \lambda_{Xe} Xe - K_3\phi Xe \end{aligned} \tag{5.8.2}$$

with symbols given in Table 5.8.1. Upon inserting appropriate values into the xenon

equation, it can readily be seen that the direct-production term $K_2\phi$ is more than one decade smaller than the iodine decay contribution. In many cases it can therefore be neglected in dynamic studies.

It is obvious that real-time simulation of posion generation and decay is somewhat impractical because of the λ values for iodine and xenon. Real-time simulation would require the use of very special servo elements and highly stabilized power supplies and reference voltages.

The dependence of xenon concentration on steady-state neutron flux is shown in relative terms in Fig. 5.8.5.

One method of simulating the poison equations is shown in Fig. 5.8.6. For simplicity, the xenon direct-generation term may be omitted. The time and amplitude scaling can be developed with conventional analog techniques (Sec. 5.1).

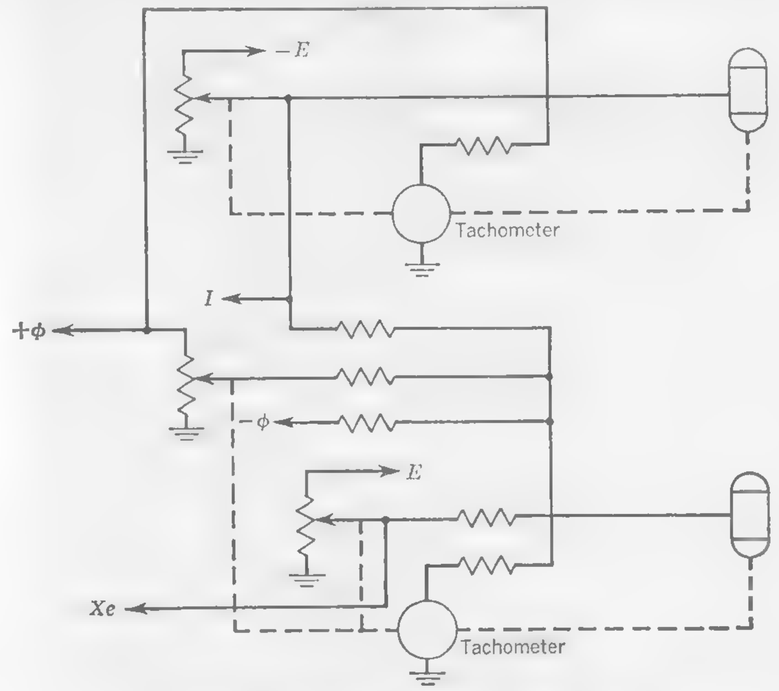


FIG. 5.8.7. Simulation of reactor poisoning.

A second method using servomultipliers is shown in Fig. 5.8.7. With certain analog systems this method may have advantages when operated with slower time scales.

5.8.4. Representation of Heat Transfer (see also Secs. 5.7 and 5.10). If the power transfer in the core and external apparatus is simulated within small transient limits, a linear solution will usually provide acceptable accuracy for a first-order approximation. Large-scale transients greatly increase the error of indicated response because of the unconsidered changes in thermodynamic and, in some cases, hydrodynamic parameters. Simulation including the nonlinear and second-order variations involves a large amount of computer equipment.

Power transfer between a metal fuel element and a liquid coolant flowing over the surface is described by the differential equation

$$P = (mc)_f \frac{d\bar{T}_f}{dt} + K_f(\bar{T}_f - \bar{T}_c) \tag{5.8.3}$$

if one neglects coefficient variations as a function of the particular temperatures.

The power absorbed by the coolant and transferred to external heat exchangers or other apparatus satisfies the relation

$$K_f(\bar{T}_f - \bar{T}_c) = (mc)_c \frac{d\bar{T}_c}{dt} + (mc)_c(T_o - T_i) \tag{5.8.4}$$

Once again, the variation of thermodynamic properties with temperature is neglected. The change in coolant temperature in passing through the core is described by

$$T_o = T_i + \frac{K_f(\bar{T}_f - \bar{T}_c)\tau_c}{(mc)_c\omega} \tag{5.8.5}$$

The parameters in Eqs. (3) to (5) are defined in Table 5.8.1.

Using these equations, the simulation of core heat transfer on an average basis could be accomplished in the manner of Fig. 5.8.8. This computer setup represents the

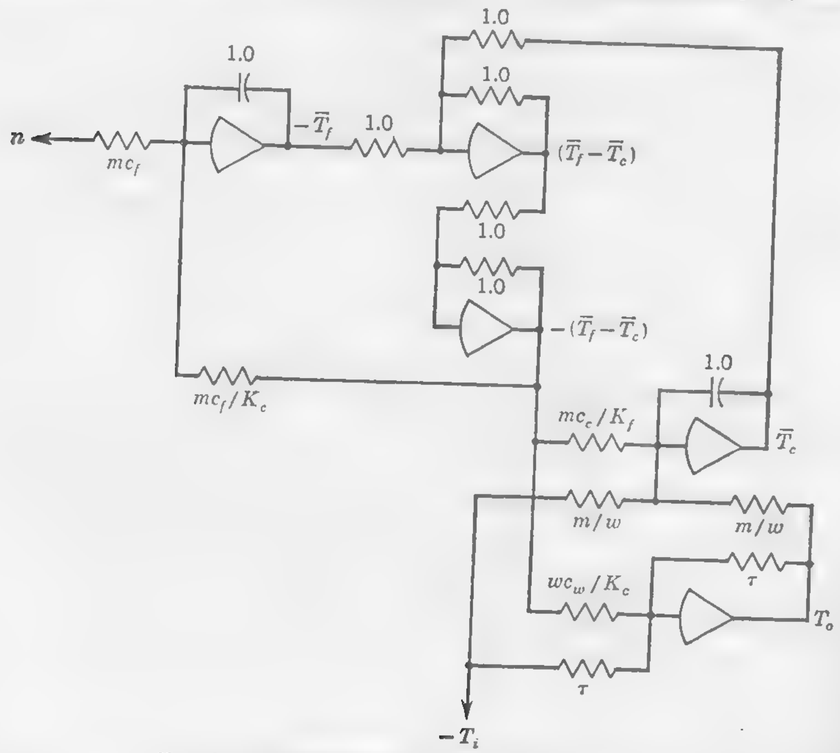


FIG. 5.8.8. Simplified heat-transfer simulation.

average heat-transfer characteristics of a core without consideration of the parameter variation as a function of temperature. Additional servomultipliers and function generators would be required to account for these nonlinear factors in the simulation. The variation of certain heat-transfer parameters with temperature, mass-flow rate, and pressure is discussed in Refs. 1 and 2.

The following equations describe the average heat transfer in a second exchanger. The general form will apply to any coolant with proper adjustment of parameters (see also Sec. 5.7):

$$\begin{aligned}
(wc)_c(T_{ix} - T_{ox}) &= K_{cx}(\bar{T}_c - \bar{T}_m) - (mc)_c \frac{d\bar{T}_c}{dt} \\
K_{cx}(\bar{T}_c - \bar{T}_m) &= (mc)_m \frac{d\bar{T}_m}{dt} + K_{ms}(\bar{T}_m - \bar{T}_s) \\
K_{ms}(\bar{T}_m - \bar{T}_s) &= (mc)_s \frac{d\bar{T}_s}{dt} + M'
\end{aligned} \tag{5.8.6}$$

These equations describe the heat balance between the primary coolant, primary piping, and secondary coolant. In many cases the metal temperature for heat-exchanger primary coolant-piping term can be neglected, since the time constant is usually comparatively small. The simulation could then be simplified by the elimination of the primary metal transfer function.

As mentioned previously, spatial variations must be considered in some cases, particularly with reactors having relatively large core mass and coolant volumes. These considerations will increase the accuracy of representation, especially for the higher-frequency responses of the system. The individual temperature coefficient of reactivity feedback terms will show the effects of system high-frequency components in the stabilizing of operating power, assuming that this feedback term is negative.

A complete block diagram of a power reactor with temperature coefficient of reactivity and poison-feedback terms included is shown in Fig. 5.8.9; the system is represented on an average-lumped-parameter basis.

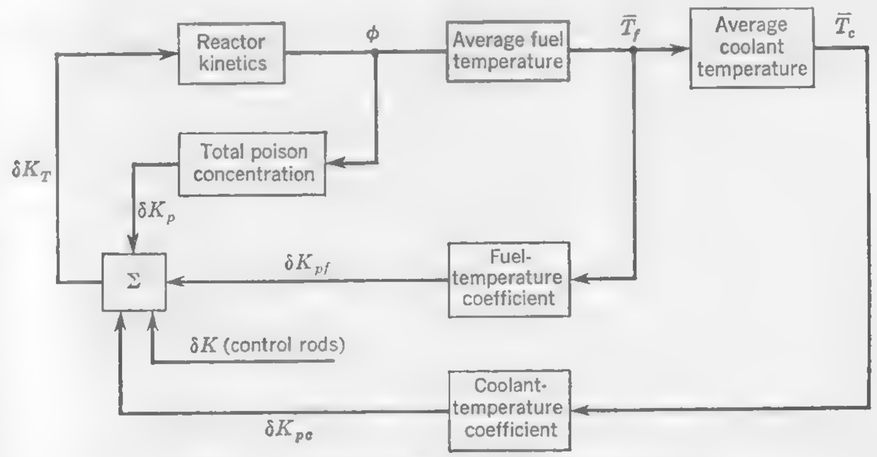


FIG. 5.8.9. Block diagram for a power reactor.

Plenum chambers on reactor vessels and heat exchangers can be represented as first-order lag terms since these are essentially mixing chambers (see also Sec. 5.7). Under conditions of low Reynolds number in coolant piping and/or short pipe runs, the transport delay can again be simulated by a first-order lag. With a high Reynolds number and a corresponding high velocity a special *time-delay unit* (Sec. 6.1) should be used.

References

1. McAdams, W. H.: *Heat Transmission*, 3d ed., McGraw-Hill Book Company, Inc., New York, 1954.
2. Behnltz, M. A.: *Control of Nuclear Reactors and Power Plants*, McGraw-Hill Book Company, Inc., New York, 1955.
3. McAdams, W. H., *et al.*: Heat Transfer at High Rates to Water with Surface Boiling, ANL–4268, December, 1948.
4. Bitone, J. J., and E. R. Mann: Nuclear Reactor Simulator, ORNL—1632.
5. Glasstone, S., and M. Edlund: *The Elements of Nuclear Reactor Theory*, D. Van Nostrand Company, Inc., Princeton, N.J., 1952.
6. U.S. Atomic Energy Commission: *The Reactor Handbook*, vol. 2, May, 1955.
7. Bell, P. R., and H. Strauss: Electronic Pile Simulator, *Review of Sci. Instruments*, August, 1950.

PART 9: RANDOM-PROCESS STUDIES

By VINCENT C. RIDEOUT

BASIC DEFINITIONS AND FORMULAS

5.9.1. Introduction. Articles 5.9.2 to 5.9.8, adapted from Ref. 30, briefly list the most important definitions and formulas required in random-process studies. Basic probability theory, mathematical statistics, and random-process theory are treated in standard texts.[1 to 7]

5.9.2. Description of Random Processes. A real *random process* is generated by random selection of a real *sample function* $x(t)$ from a population or *ensemble* of such functions. The joint realization of any finite set of *sample* values $x(t_1)$, $x(t_2)$, . . . , $x(t_n)$ must be a random event permitting definition of a joint probability distribution (*nth probability distribution* of the random process for the set of *sampling times* t_1, t_2, . . . , t_n). In particular, the *first* and *second probability densities* are

$$p_1(X_1,t_1) = \frac{\partial}{\partial X_1}\,\text{Prob}\,[x(t_1) \leq X_1] \tag{5.9.1}$$

$$p_2(X_1,t_1;\,X_2,t_2) = \frac{\partial^2}{\partial X_1\,\partial X_2}\,\text{Prob}\,[x(t_1) \leq X_1,\, x(t_2) \leq X_2] \tag{5.9.2}$$

A random process is *stationary* if and only if all its probability distributions remain unchanged if t_k is replaced by $t_k + a$ (shift of time origin). A random process is *Gaussian* if and only if all its probability distributions are normal distributions.

Two random processes described by $x(t)$, $y(t)$ can have *joint probability distributions* of sets of sample values $x(t_i)$, $y(t_k)$. Two random processes may or may not be *jointly stationary* and/or *jointly Gaussian*.

5.9.3. Ensemble Averages and Correlation Functions. The *ensemble average* of a given function $f = f[x(t_1),x(t_2),\ \ldots\ ,x(t_n)]$ of n sample values (*statistic*) is its mathematical expectation $E\{f\}$ over their joint distribution,

$$E\{f\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f p_n(x_1,t_1;\,x_2,t_2;\ \ldots\ ;\,x_n,t_n)\,dx_1\,dx_2\,\cdots\,dx_n \tag{5.9.3}$$

and is thus a function of the sampling times t_1, t_2, . . . , t_n. Suitable limiting procedures can also define ensemble averages of *integrals* like $\int_a^b x(t)\,dt$.

The most important ensemble averages are $E\{x(t_1)\}$ (*expected value* or *first-order moment* of x), $E\{[x(t_1)]^2\}$ (*mean square* or *second-order moment* of x), and

$$E\{x(t_1)x(t_2)\} = \phi_{xx}(t_1,t_2) = \phi_{xx}(t_2,t_1) \quad \text{(ENSEMBLE AUTOCORRELATION FUNCTION OF } x\text{)} \tag{5.9.4}$$

and, in terms of the joint distribution of $x(t)$ and $y(t)$

$$E\{x(t_1)y(t_2)\} = \phi_{xy}(t_1,t_2) = \phi_{yx}(t_2,t_1) \quad \text{(ENSEMBLE CROSS-CORRELATION FUNCTION OF } x \text{ AND } y\text{)} \tag{5.9.5}$$

$$E\{[x(t_1)-y(t_2)]^2\} = \phi_{xx}(t_1,t_2) + \phi_{yy}(t_1,t_2) - 2\phi_{xy}(t_1,t_2) = \epsilon_{xy}{}^2(t_1,t_2) \quad \text{(ENSEMBLE MEAN-SQUARE DELAY ERROR)} \tag{5.9.6}$$

For *stationary* processes $E\{x(t_1)\}$ and $E\{[x(t_1)]^2\}$ are constant, and correlation function and delay error reduce to functions $\phi_{xx}(\tau)$, $\phi_{xy}(\tau)$, $\epsilon_{xy}{}^2(\tau)$ of the *delay* $\tau = t_2 - t_1$.

5.9.4. Effect of Linear Operations. Let $x(t)$, $y(t)$, $z(t)$ describe real random processes related so that $z(t) = \alpha x(t) + \beta y(t)$. Then

$$\begin{aligned}\phi_{xz} &= \alpha\phi_{xx} + \beta\phi_{xy}\\ \phi_{zz} &= \alpha^2\phi_{xx} + \beta^2\phi_{yy} + \alpha\beta(\phi_{xy} + \phi_{yx})\end{aligned} \tag{5.9.7}$$

For a linear (not necessarily time-invariant) system (filter, servo) with input $r(t)$, output $c(t)$, and impulse response or *weighting function* $w(t,\lambda)$

$$c(t) = \int_{-\infty}^{t} w(t,\lambda)\,r(\lambda)\,d\lambda \quad \text{(SUPERPOSITION INTEGRAL)} \tag{5.9.8}$$

$$\phi_{rc}(t_1,t_2) = \int_{-\infty}^{t_2} w(t_2,\lambda)\,\phi_{rr}(t_1,\lambda)\,d\lambda = \phi_{cr}(t_2,t_1) \quad \text{(WIENER-LEE RELATION)} \tag{5.9.9}$$

$$\phi_{cc}(t_1,t_2) = \int_{-\infty}^{t_2} w(t_2,\mu)\,\phi_{cr}(t_1,\mu)\,d\mu \tag{5.9.10}$$

5.9.5. Ergodic Processes and Time Averages. A stationary random process will be called *ergodic* (more accurately, it satisfies the *ergodic hypothesis*) if and only if every ensemble average $E\{f[x(t_1),x(t_2),\ \ldots\ ,x(t_n)]\}$ equals the corresponding *time average*

$$\langle f\rangle = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} f[x(t_1+t),\,x(t_2+t),\ \ldots\ ,\,x(t_n+t)]\,dt \tag{5.9.11}$$

with probability 1. Note that $\bar{f}$, like f, is a function of sample values (statistic), and hence a *random variable*. In the case of ergodic processes, finite-T approximations to $\bar{f}$ serve as *estimates* of the ensemble average $E\{f\}$.

5.9.6. Time-correlation Functions. In particular, the ensemble correlation functions (5), (6) of real ergodic processes are effectively replaced by the functions

$$\phi_{xx}(\tau) = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)x(t+\tau)\,dt = \phi_{xx}(-\tau) \quad \text{(AUTOCORRELATION FUNCTION)} \tag{5.9.12}$$

$$\phi_{xy}(\tau) = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)y(t+\tau)\,dt = \phi_{yx}(-\tau) \quad \text{(CROSS-CORRELATION FUNCTION)} \tag{5.9.13}$$

The limits (12) and (13) exist whenever $\langle x^2(0)\rangle = \phi_{xx}(0)$ and $\langle y^2(0)\rangle = \phi_{yy}(0)$ exist. The time-correlation functions (12), (13) satisfy the relations (7), (9), and (10) if one substitutes $t_1 = 0$, $t_2 = \tau$.

5.9.7. Correlation Functions and Spectra. The **power spectral density** $\Phi_{xx}(j\omega)$ of a real time function $x(t)$ (or of an ergodic random process) and the **cross-spectral density** $\Phi_{xy}(j\omega)$ of $x(t)$ and $y(t)$ are defined by the *Wiener-Khintchine relations*

$$\Phi_{xx}(j\omega) = \int_{-\infty}^{\infty}\phi_{xx}(\tau)e^{-j\omega\tau}\,d\tau = 2\int_0^{\infty}\phi_{xx}(\tau)\cos\omega\tau\,d\tau$$
$$\phi_{xx}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\Phi_{xx}(j\omega)e^{j\omega\tau}\,d\omega = \frac{1}{\pi}\int_0^{\infty}\Phi_{xx}(j\omega)\cos\omega\tau\,d\omega \tag{5.9.14}$$

$$\Phi_{xy}(j\omega) = \int_{-\infty}^{\infty}\phi_{xy}(\tau)e^{-j\omega\tau}\,d\tau \qquad \phi_{xy}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\Phi_{xy}(j\omega)e^{j\omega\tau}\,d\omega \tag{5.9.15}$$

The term "power spectral density" stems from the relation

$$\langle x^2(0)\rangle = \phi_{xx}(0) = \int_{-\infty}^{\infty}\Phi_{xx}(j\omega)\,\frac{d\omega}{2\pi} \tag{5.9.16}$$

5.9.8. Effect of Linear Operations on Spectra. If $x(t)$, $y(t)$, $z(t)$, $r(t)$, $c(t)$ are real power-limited time functions,

$$\Phi_{xz} = \alpha\Phi_{xx} + \beta\Phi_{xy} \qquad \Phi_{zz} = \alpha^2\Phi_{xx} + \beta^2\Phi_{yy} + \alpha\beta(\Phi_{xy} + \Phi_{yx}) \tag{5.9.17}$$

$$\Phi_{rc}(j\omega) = F(j\omega)\Phi_{rr}(j\omega) \tag{5.9.18}$$

$$\Phi_{cc}(j\omega) = |F(j\omega)|^2\Phi_{rr}(j\omega) \tag{5.9.19}$$

where $F(j\omega)$ is the frequency-response function of the given linear system.

ANALOG-COMPUTER TECHNIQUES: INTRODUCTION

5.9.9. Analog Computation of Statistical Quantities: Survey. The end products of all analog-computer random-process studies are estimated or inferred values of *ensemble* statistics (ensemble averages, mean-square errors, maximum and minimum errors, correlation functions, spectra, probabilities) obtained (1) by computation of corresponding time averages, using a real or simulated system with random inputs (ergodic processes only), or (2) by mathematical operations on related known, estimated, or inferred ensemble statistics (e.g., adjoint method, Art. 5.9.16), or (3) by empirical ensemble averaging, using a real or simulated system with random inputs ("Monte Carlo" method, especially effective with fast repetitive computers, Sec. 6.3).

Analog *spectrum analyzers and correlators* are often built up largely from standard computing elements; or, if built commercially as special-purpose devices, they can often be incorporated into general-purpose analog computers.

Equipment for *correlation and power-spectrum measurements* (Arts. 5.9.12 to 5.9.21) must be matched to the spectral range of the random-process data to be studied. Correlation functions and spectra may be determined from:

1. The original source; this often requires a multichannel "parallel" correlator or spectrum analyzer.
2. A magnetic-tape record of the original data, permitting repetitive playback and use of a "serial" correlator or spectrum analyzer. Playback speed may be increased to permit faster processing.
3. An analog-computer simulation of the system with random inputs. A time-scale change can make processing faster (or slower) than in the original system.

Similar frequency-content and sampling-time considerations apply to the design of devices for analyzing the *amplitude distribution* of a random process (Arts. 5.9.22 and 5.9.23). In addition, prevention of overload due to random peaks requires careful scaling (Art. 5.9.11).

5.9.10. Noise Generators and Spectrum Shaping. Some of the computer methods to be discussed require *Gaussian-noise generators* (Sec. 3.3) whose spectral density is flat ("white" noise) over a band so wide that the system under study rather than the noise generator determines the output spectrum.

To produce Gaussian noise with a desired power spectral density $\Phi(\omega)$, one passes white noise through a linear spectrum-shaping filter whose frequency-response function $F(j\omega)$ permits one to realize or approximate $\Phi(j\omega) = \text{const}\ |F(j\omega)|^2$. A method for fitting the product

$$F(p) = \frac{a_0 + b_0 p}{d_0 + e_0 p + f_0 p^2} \cdot \prod_{n=1}^{N-1} \frac{a_n + b_n p + c_n p^2}{d_n + e_n p + f_n p^2}$$

to a power spectrum with a small number of peaks has been discussed by Mazelsky and Amey.[8]

5.9.11. Scaling Problems and Prewhitening.[9] Many noise distributions, including the Gaussian, have a large ratio of peak amplitude to rms amplitude. In order to steer a middle course between overload and chopper-noise problems in the analog computer, it is necessary to scale with care. The scaling problem is intensified in spectrum analysis with narrow-band tunable filters if the spectrum contains large peaks which may obscure results elsewhere in the spectrum because of "leakage" in the skirts or minor lobes of the filters. The term *prewhitening* refers to the use of an added filter at the input, with a frequency response which will make the output, which is to be measured, more nearly of constant-power spectral density. After the spectrum is determined, a correction is made for the prewhitening filter; *this scheme is restricted to linear-system studies with Gaussian inputs.*

DETERMINATION OF CORRELATION FUNCTIONS AND DELAY ERROR

5.9.12. Analog Measurement of Correlation Functions and Delay Error: Stationary Case. Straightforward schemes for measuring time-correlation functions require

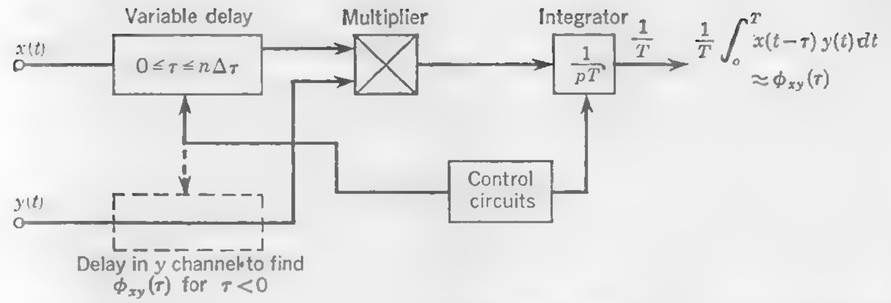


Fig. 5.9.1*a*. A simple serial correlator using a variable delay line. The control circuit resets the integrator to zero after T sec and increases the delay by $\Delta\tau$. $y(t) \equiv x(t)$ for autocorrelation.

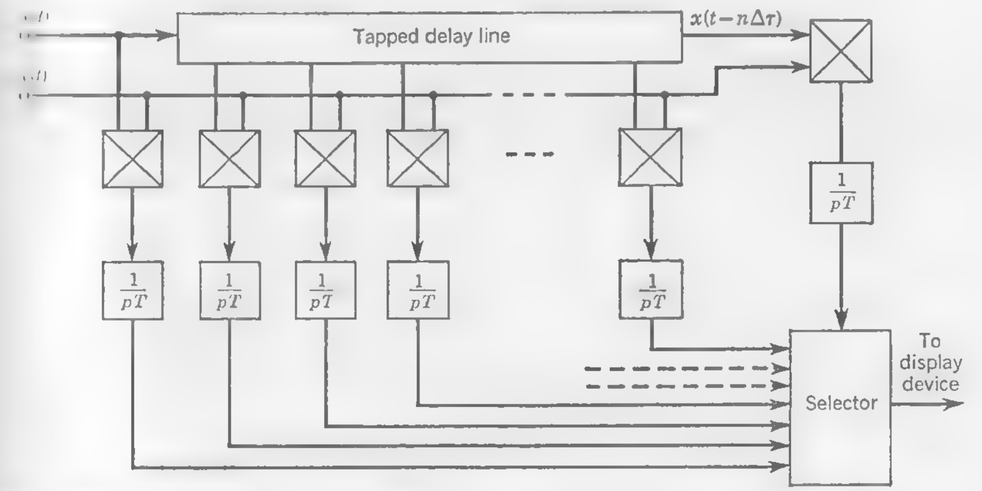


Fig. 5.9.1*b*. A parallel correlator using a tapped delay line. This device will determine $\phi_{xy}(\tau)$ for $\tau \geq 0$ at $n+1$ steps of $\Delta\tau$. Values for $\tau < 0$ can be obtained by reversing inputs, or by use of twice as much equipment as shown.

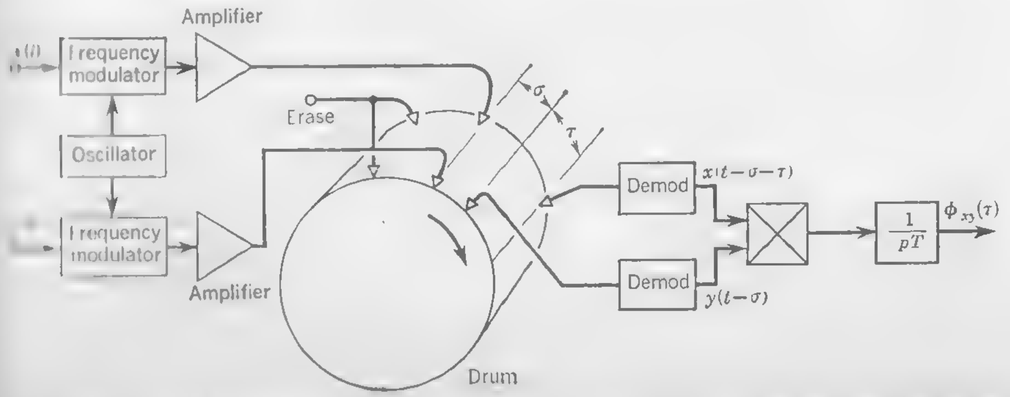


Fig. 5.9.2. Drum correlator using frequency modulation to minimize recorder distortion (see also Sec. 6.1).

delay, multiplication, and integration. There are two common methods of delaying functions, viz., tapped passive delay lines[11] (Fig. 5.9.1), and magnetic-tape or drum[12] systems with movable heads (Fig. 5.9.2; see also Sec. 6.1). In Fig. 5.9.1*b*, a multiplicity of multipliers and integrators are connected to taps on the delay line, so that, at considerable cost in equipment, this machine will determine n points of a correlation curve n times as fast as the simple correlator of Fig. 5.9.1*a*.

The magnetic-drum delay scheme of Fig. 5.9.2 permits delays up to nearly the time of one-half of a drum revolution with one movable reading head, or twice this with two movable reading heads. Tape may also be used, and either the reading head may be moved, or a loop of tape between reading and writing head may be changed in length (note, however, the accuracy limitations discussed in Sec. 6.1). Tape and drum systems have the advantage that low-frequency phenomena may be recorded slowly, and then speeded up for processing, if desired.

Closely related to correlation measurement is the measurement of the *average squared delay error*[13]

$$E(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_0^T [r(t) - c(t+\tau)]^2\,dt = \phi_{rr}(0) + \phi_{cc}(0) - 2\phi_{rc}(\tau) \qquad (5.9.20)$$

which is an important measure of system performance (Fig. 5.9.3).[13] Note that it is as easy to examine averages of other functions; thus, substitution of an absolute-value unit in place of the multiplier in Fig. 5.9.3 yields the average absolute delay error $\langle|r(t) - c(t+\tau)|\rangle$

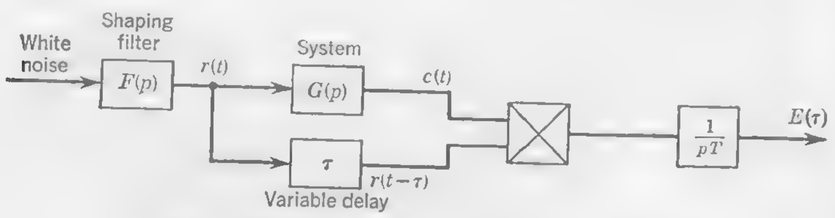


FIG. 5.9.3. Determination of mean-square delay error.

Other more indirect methods of correlation measurement may offer certain advantages. Thus Lampard[14] uses filter systems whose impulse responses are members of a Laguerre system together with a multiplier for the determination of autocorrelation functions.

5.9.13. Errors in Time-correlation Measurements. Among these, the most important types of errors afflicting time-correlation measurements are *errors originating in the analog equipment* (particularly the multiplier) and *error due to finite integration time.* Computation problems are intensified by the wide amplitude range often encountered with random signals. High-speed processing requires wide-band multipliers, which again tend to compromise accuracy.

Finite-integration-time error resulting from data or computer limitations is of a more fundamental and serious nature. Bendat and others[5] have examined the random error in the autocorrelation function which results from the use of a single piece of data of finite length T. His results *for a simple exponential correlation curve* $\phi(\tau) = e^{-b|\tau|}$ show, for example, that at $b|\tau| = 1$, or $\phi(\tau) = 1/e$, one requires $bT = 6{,}000$ to be 95 per cent certain that the value of the correlation is correct within ± 10 per cent. Note that this means that $T/|\tau| = 6{,}000$; so that, if $\tau = 1$ sec, $T = 6{,}000$ sec, or 1?[illegible] hr. This is prohibitively long if a serial correlator is to be used and may even be too long for a parallel correlator; note the advantage of using a fast time scale.

5.9.14. "Mathematical" Computation of Correlation Functions and Delay Error from Known Correlation Functions: Stationary Case. In the case of stationary ergodic processes and time-invariant linear systems, one can replace the input-output relations (8) to (10) by

$$c(t) = \int_{-\infty}^{t} h(t-\lambda)\,r(\lambda)\,d\lambda \qquad (5.9.21)$$

$$\phi_{rc}(\tau) = \int_{-\infty}^{\tau} h(\tau-\lambda)\phi_{rr}(\lambda)\,d\lambda = \phi_{cr}(-\tau) \qquad (5.9.22)$$

$$\phi_{cc}(\tau) = \int_{-\infty}^{\tau} h(\tau-\mu)\phi_{cr}(\mu)\,d\mu \qquad (5.9.23)$$

where $h(t-\lambda) \equiv w(t,\lambda)$. Comparison of Eqs. (22) and (23) with the linear-system-response expression (21) indicates the possibility of computing $\phi_{rc}(\tau)$, $\phi_{cc}(\tau)$, and hence

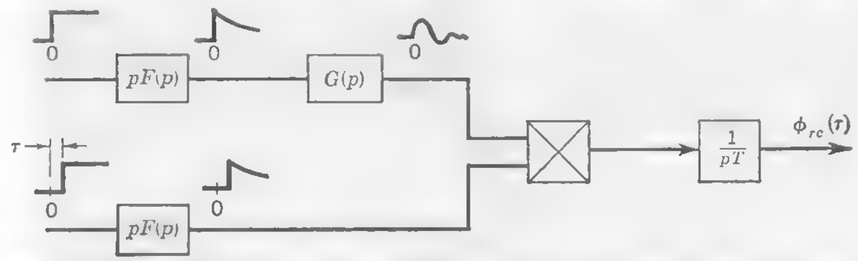


FIG. 5.9.4. Analog method for determining the input-output cross-correlation function of a linear system $G(p)$ for a white Gaussian input shaped by a linear network $F(p)$. Note the use of $pF(p)$ to permit the use of step inputs rather than δ functions. This method can be used to determine $\phi_{cc}(\tau)$ if a second simulated system described by $G(p)$ is introduced at the output of the lower filter. If only $\phi_{cc}(0) = E\{c^2(t)\}$ is required, a single setup of $G(p)$ and $F(p)$ suffices. Suitable initial conditions can often replace the step input.

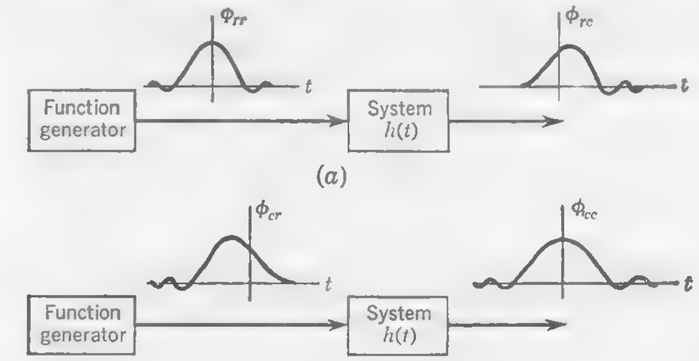


FIG. 5.9.5*a*. An input voltage $\phi_{rr}(\tau)$ applied to a system of given impulse response $h(t)$ will yield the output $\phi_{rc}(\tau)$, where τ is represented by the computer time.
FIG. 5.9.5*b*. Once $\phi_{rc}(\tau)$ has been determined, $\phi_{rc}(-\tau) = \phi_{cr}(\tau)$ can be generated by a function generator and applied to the given system to yield $\phi_{cc}(\tau)$.

also $E(\tau)$, by the methods of Figs. 5.9.4 and 5.9.5.[15 to 17] Note that τ is now represented by the computer time. In the frequent cases where the correlation functions die away to zero quickly,[5] or if "lag windows" (Art. 5.9.21) can be used, this approach saves much time compared with random-input simulation.

5.9.15. System Analysis for Nonstationary Problems. For nonstationary random processes, the system output or error $c(t)$ for a single input $r(t)$ is given by Eq. (8), and $\phi_{rr}(t_1,t_2)$ and $\phi_{cc}(t_1,t_2)$ are given by Eqs. (9) and (10). The following situations arise in practice.

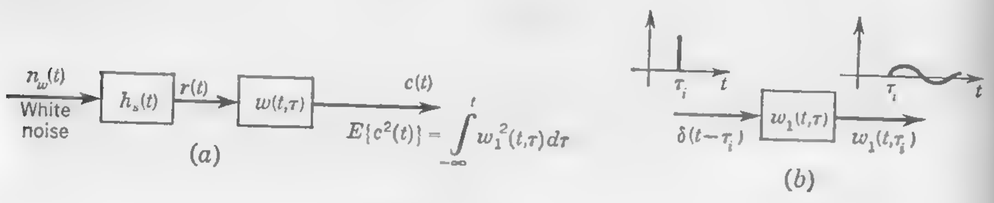
Case 1. In many applications, $r(t) = 0$ for $t \le 0$, so that the lower limits in Eqs. (9) and (10) are zero, and one wishes to determine the mean-square output or error

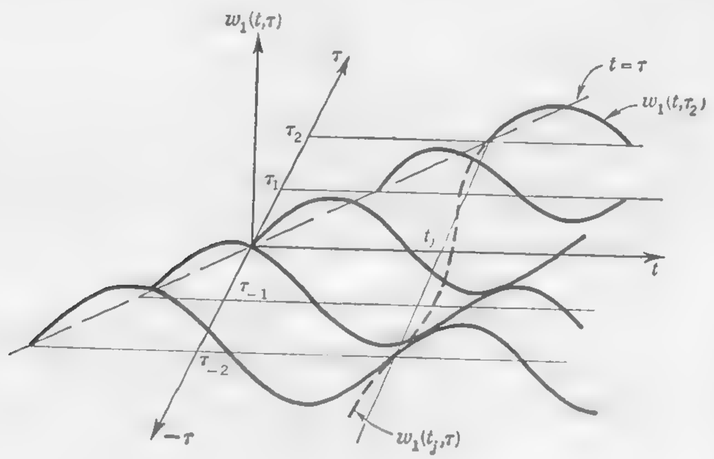
$$E\{c^2(t_1)\} = \phi_{cc}(t_1,t_1) = \int_0^{t_1} w(t_1,\mu)\phi_{cr}(t_1,\mu)d\mu \qquad (5.9.24)$$

where

$$\phi_{cr}(t,\mu) = \int_0^{t} w(t,\lambda)\phi_{rr}(\mu,\lambda)d\lambda \qquad (5.9.25)$$

is the response of the given time-varying system to the input $\phi_{rr}(\mu,t)$. $\phi_{cr}(t_1,\mu)$ can thus be obtained as a function of μ through successive computer runs with inputs $\phi_{rr}(\mu,t)$ with different values of μ. $\phi_{cr}(t_1,\mu)$ can then be generated in a function generator as a function of μ, which is now represented by the computer time. Application of this time function to the simulated system yields $E\{c^2(t_1)\}$ for various selected values of t_1 without any need for explicit analytical formulation of $\phi_{rr}(t_1,t_2)$ or $w(t,\lambda)$; but the process is a tedious one.





(c)

FIG. 5.9.6*a*. A shaping filter $h_s(\lambda)$ with white-noise input yields the prescribed stationary input to the given time-varying system $w(t,\tau)$. The combined system, described by the weighting function $w_1(t,\tau)$, has the correct output autocorrelation function for white-noise input.

FIG. 5.9.6*b*. Simulation of the combined system yields its impulse response $w_1(t,\tau)$ for fixed values of $\tau = \tau_i$, and $w_1(t_j,\tau)$ is determined by cross-plotting from these curves. Then $E\{c^2(t)\}$ is found from Eq. (27) through numerical integration with respect to τ.

FIG. 5.9.6*c*. Graphical determination of $w_1(t_j,\tau)$ from impulse responses $w_1(t,\tau_i)$ (cross plotting).

Case 2. If the input $r(t)$ is a stationary random process, then the mean-square output or mean-square error is given by

$$E\{c^2(t_1)\} = \int_{-\infty}^{t_1} d\mu\, w(t_1,\mu) \int_{-\infty}^{t_1} w(t_1,\lambda)\phi_{rr}(\lambda - \mu)d\lambda \tag{5.9.26}$$

To simplify this expression, one introduces a shaping filter whose weighting function $h_S(t,\lambda)$ yields the desired output correlation function $\phi_{rr}(t)$ for white-noise input. This shaping filter is combined with the given system to form a new system described by $w_1(t,\tau)$ (Fig. 5.9.6*a*). The input autocorrelation function for this new system is $\phi'_{rr}(\tau) = \delta(\tau)$, so that

$$E\{c^2(t_1)\} = \int_{-\infty}^{t_1} w_1^2(t_1,\tau)\, d\tau \tag{5.9.27}$$

This again points the way to a solution without random input and ensemble averaging (Fig. 5.9.6*b*,*c*). Direct use of this scheme is still a fairly tedious process despite its simplicity and the reduced integration times possible, since repeated runs are required for different values of τ. As seen in Fig. 5.9.6*b* and *c*, the simulator output is a function of t (computer time), whereas Eq. (27) requires integration with respect to τ.

5.9.16. The Adjoint-system Method. The *adjoint-system method* of Laning and Battin[4] avoids the necessity of repeating runs for different values of τ. For a given

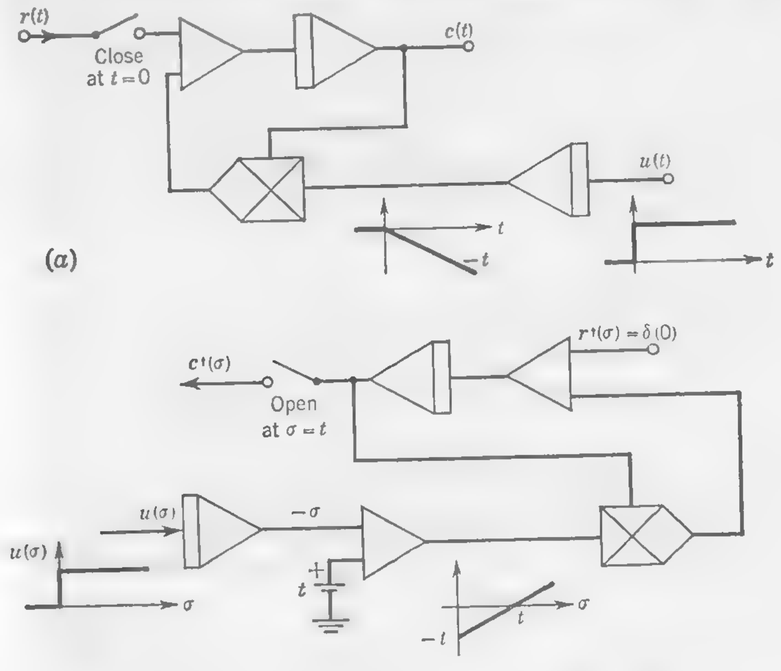


FIG. 5.9.7*a*. Direct analog simulation of the time-varying system defined by

$$\frac{dc(t)}{dt} + tc(t) = r(t)$$

where $r(t)$ is white noise applied at $t = 0$, and initial conditions are zero.

FIG. 5.9.7*b*. Adjoint system, corresponding to

$$\frac{dc\dagger}{dt} + (t_j - \sigma)c\dagger = r\dagger(\sigma)$$

Using this adjoint system, one can find $E\{c^2(t)\}$ for each value $t = t_1$ of interest by squaring and integrating $c\dagger(\sigma)$ for unit-impulse input over the period $0 < \sigma \leq t_1$.

linear time-variable system with impulse response $w(t,\tau)$, one constructs the *adjoint system* whose response at the time t_1 to a unit impulse at the time τ is given by

$$w\dagger(t_1,\tau) \equiv w(\tau,t_1) \tag{5.9.28}$$

so that, for a system with white-noise input,

$$E\{c^2(t_1)\} = \int_{-\infty}^{t_1} w^2(t_1,\tau)\, d\tau = \int_{-\infty}^{t_1} w\dagger^2(\tau,t_1)\, d\tau$$

Since a system with the unit-impulse response $w\dagger(t, \lambda)$ is not physically realizable, one introduces a new time variable $\sigma = t_1 - \tau$, so that

$$E\{c^2(t_1)\} = \int_0^{\infty} w\dagger^2(t_1 - \sigma,t_1)\, d\sigma \tag{5.9.29}$$

For a given system, which is assumed to have one input and one output, one constructs the adjoint system as follows:

1. Rewrite the block diagram of the original system so that all amplifiers have unit input gains, with gains greater as well as less than unity temporarily incorporated into fixed or time-variable potentiometer settings, and so that no potentiometer has more than one output (repeat potentiometers where necessary).
2. Reverse the direction of signal flow through each computing element.

Next one introduces $\sigma = t_1 - \tau$ as the computer time through the substitution $\tau = t_1 - \sigma$ for the problem time wherever it appears (potentiometer drives, switch timing, etc.). Now one applies a unit impulse at the physical time $\sigma = 0$ and integrates the squared system output $w^2(t_1 - \sigma, t_1)$ for a sufficiently long time to obtain $E\{c^2(t_1)\}$ *in one computer run for each value of* t_1. All initial values are taken to be zero.

Figure 5.9.7 shows an example of adjoint-system construction. For system inputs other than white noise, one would have to add a shaping filter to the original system and then construct the adjoint of the combined system.

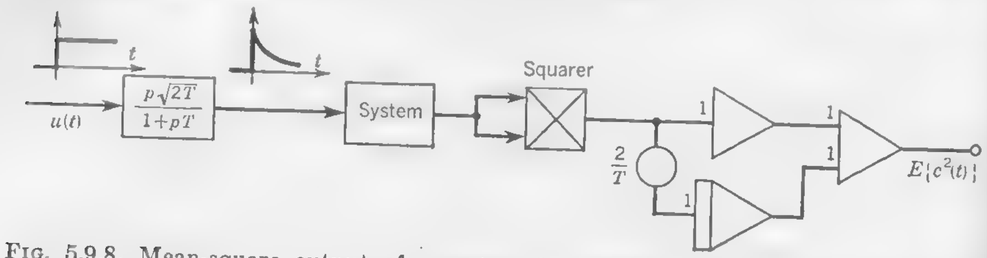


FIG. 5.9.8. Mean-square output of a system for suddenly applied stationary random input $\phi_{rr}(\tau) = e^{-\tau/T}$. As shown in Eq. (11), this may be simulated with the aid of the shaping network, $\sqrt{2T}/(1 + pT)$, corresponding to ϕ_{rr}. The derivative operator included above permits a step rather than an impulse input, and may be included in the simulation of the system.

5.9.17. A Special Case. *If the system has constant coefficients, except for a switch which applies a stationary input at* $t = 0$, then this switch is the only time-varying parameter. Although this case can be solved by the adjoint method, it may be more simply handled if the input autocorrelation function is of a simple known form. In this case, $h(t - \lambda)$ replaces $w(t,\lambda)$ in Eq. (26), and the lower limits on the integrals are zero.

For the example of an input consisting of white noise filtered by a simple RC network

$$\phi_{rr}(\tau) = e^{-|\tau|/T} \qquad \Phi_{rr}(\omega) = \frac{2T}{\omega^2 T^2 + 1}$$

and Eq. (26) yields

$$E\{c^2(t_1)\} = \int_0^{t_1} d\mu\, h(t_1 - \mu) \int_0^{t_1} h(t_1 - \lambda) e^{-|\lambda-\mu|/T}\, d\lambda$$

$$= F^2(t_1) + \frac{2}{T}\int_0^{t_1} F^2(t)\, dt \qquad (5.9.30)$$

where

$$F(t) = \int_0^{t_1} h(t - \sigma) e^{-\sigma/T}\, d\sigma \qquad (5.9.31)$$

$E\{c^2(t_1)\}$ can, thus, be computed without the use of random inputs (Fig. 5.9.8).

MEASUREMENT OF SPECTRAL DENSITIES

5.9.18. Measurement of Power Spectra. The power-density-spectrum estimate of a random signal may be determined from its autocorrelation function with the aid of Eq. (14). This method is particularly advantageous for digital computation.[18] With analog techniques, it usually seems best to measure the power spectrum directly, if it is of primary interest. Cross-spectral densities, on the other hand, are complex functions and hence complicated to measure; here Fourier inversion of cross-correlation functions may be simpler (Sec. 5.9.21).

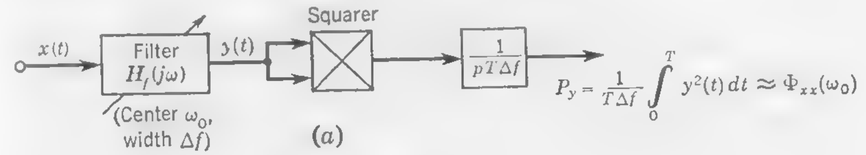
The most straightforward method for analog determination of power spectra involves a tunable narrow-band filter (Fig. 5.9.9*a*). Where possible, the signal may be changed in time scale by changes of the recorder speed[19,20] or the simulation time scale.

If the filter frequency-response function $H_f(j\omega)$ yields an ideal flat-topped pass band extending from f_1 to f_2, and if integration times are long, then the best result one can expect is an estimate

$$P_y = \int_{-\infty}^{\infty} |H_f(j\omega)|^2 \Phi_{xx}(j\omega)\, df = \int_{f_1}^{f_2} \Phi_{xx}(j\omega)\, df \qquad (\omega = 2\pi f) \qquad (5.9.32)$$

of $\Phi_{xx}(j\omega)$ at the center frequency of the filter.

Variances and confidence intervals associated with such estimates will be reduced if the spacing $\frac{1}{2}T$ of the signal Fourier components is small compared with the filter band $\Delta f = f_2 - f_1$.



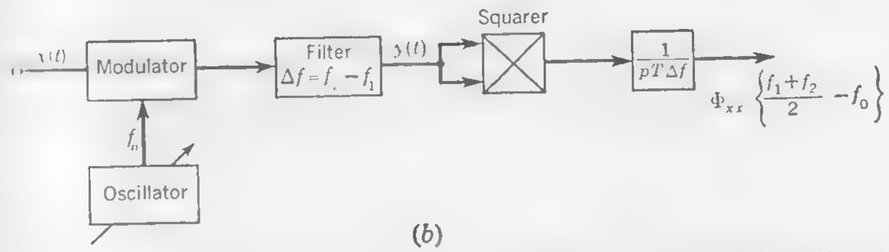


FIG. 5.9.9*a*. Spectrum analyzer using a tunable narrow-band filter $H_f(j\omega)$.
FIG. 5.9.9*b*. Heterodyne-type spectrum analyzer using a tunable beating oscillator.

To determine the power-spectral-density estimate for the given sample $x(t)$ it is necessary to tune the filter of Fig. 5.9.9*a*. This is difficult, particularly if the filter is not of the simple second-order variety; so that the heterodyne scheme of Fig. 5.9.9*b* is ordinarily used. Here the filter is set at a fixed intermediate frequency, and a beat-frequency oscillator is tuned through a range continuously or in steps. The filter may be especially designed for flat top and steep sides; its phase-shift characteristics are not particularly important, as long as the filter settling time does not become comparable with the sample length T.

In commercial spectrum analyzers a square-law detector (or even a linear detector) and RC filter may be used in place of the multiplier and integrator shown; but an analog multiplier or squaring circuit of suitable bandwidth and an integrator, which is read and reset to zero every T sec, will give more accurate results.

Spectral analysis can be greatly speeded up (at the expense of added equipment) if the parallel scheme of Fig. 5.9.10*a* is used. A parallel spectral analyzer using high-pass filters[21] is shown in Fig. 5.9.10*b*. This design gives some advantages in filter design and is claimed to reduce filter-transient errors.

5.9.19. Measurement of Cross-spectral Density.[22] The cross-spectral density of two stationary random processes is normally complex and may be written as

$$\Phi_{xy}(j\omega) = |\Phi_{xy}| e^{+j\theta_{xy}} = U_{xy}(\omega) + jV_{xy}(\omega) \qquad (5.9.33)$$

where $U_{xy}(\omega) = \int_{-\infty}^{\infty} \phi_{xy}(\tau) \cos \omega\tau\, d\tau \qquad V_{xy}(\omega) = -\int_{-\infty}^{\infty} \phi_{xy}(\tau) \sin \omega\tau\, d\tau$

If two signals $x_1(t)$, $y_1(t)$ are passed through identical narrow-band filters $H(j\omega)$, and one of them is also passed through a phase shifter $H_s(j\omega)$ (Fig. 5.9.11*a*), then from

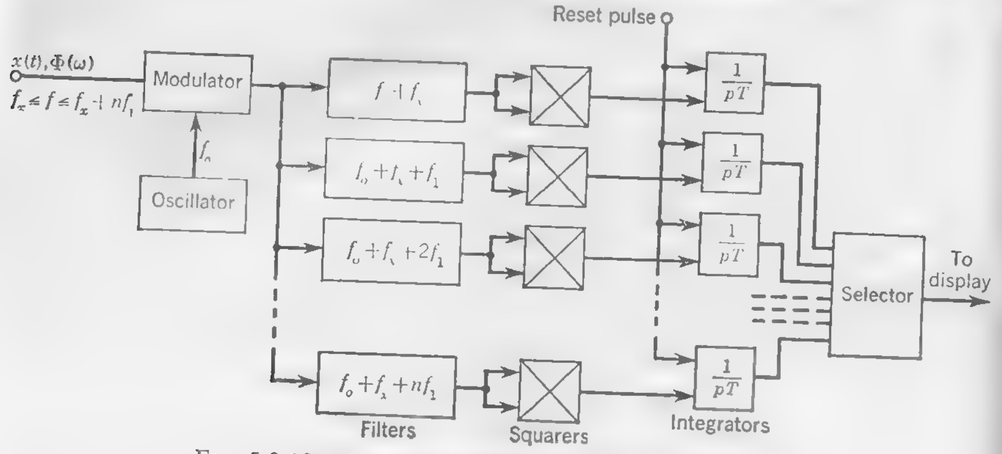


FIG. 5.9.10*a*. A parallel filter-type spectrum analyzer.

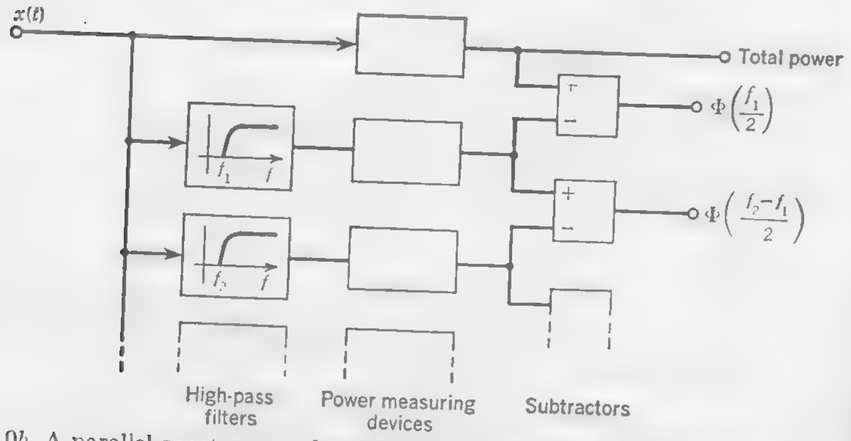


FIG. 5.9.10*b*. A parallel spectrum analyzer using high-pass filters. Subtraction yields the power in each frequency band.

Eqs. (18) and (19)

$$\Phi_{x_2y_2} = |H|^2 H_s \Phi_{x_1y_1} \tag{5.9.34}$$

If the output product $x_2(t)y_2(t)$ is averaged over a long period, the output will be

$$\frac{1}{T}\int_0^T x_2(t)y_2(t)\,dt \approx \phi_{x_2y_2}(0) = \int_{-\infty}^{\infty} \Phi_{x_2y_2}(j\omega)\,df = \int_{-\infty}^{\infty} |H|^2 H_s \Phi_{x_1y_1}(j\omega)\,df$$

$$= \begin{cases} \int_{f_1}^{f_2} U_{xy}(\omega)\,df & \text{if } H_s = 1 \\ \int_{f_1}^{f_2} V_{xy}(\omega)\,df & \text{if } H_s = j \end{cases} \tag{5.9.35}$$

provided that $H(j\omega)$ is an ideal filter of bandwidth $\Delta f = f_2 - f_1$ (Fig. 5.9.11*b*). Accurately matched filters are required.

5.9.20. Errors in Power-spectrum Measurements. Sources of error in measurements of power spectra by the methods discussed in the preceding article include

1. Errors in recording and reproducing the original data
2. Errors due to finite length of sample
3. Errors resulting from the shape of the filter characteristic
4. Errors caused by inaccuracies in the multiplier, integrator, and modulator

Of these, 1 and 4 are typical analog-computer problems and will not be dwelt upon further; in this connection, note again that random signals may easily give rise to overload problems.

The third source of error is peculiar to the filter type of spectrum analyzer. Practical filters do not have perfectly steep sides, and large peaks in the spectrum may affect readings at some low-level part because of the "skirt" response of the filter. Prewhitening (Art. 5.9.11) and multisection filters with large skirt attenuation will help.

The most fundamental limitation on the accuracy of spectral measurement results from the necessarily finite length of the record. This may be present because of some practical limitations of recording equipment, because the process is only approximately stationary, or merely because some limit must be placed on the time needed to determine the spectrum.

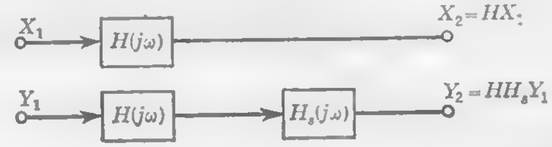


FIG. 5.9.11*a*. A cross-spectrum analyzer requires identical narrow-band filters $H(j\omega)$ in each channel, and for the quadrature component an added 90-deg phase shifter, $H_s(j\omega)$, in one channel.

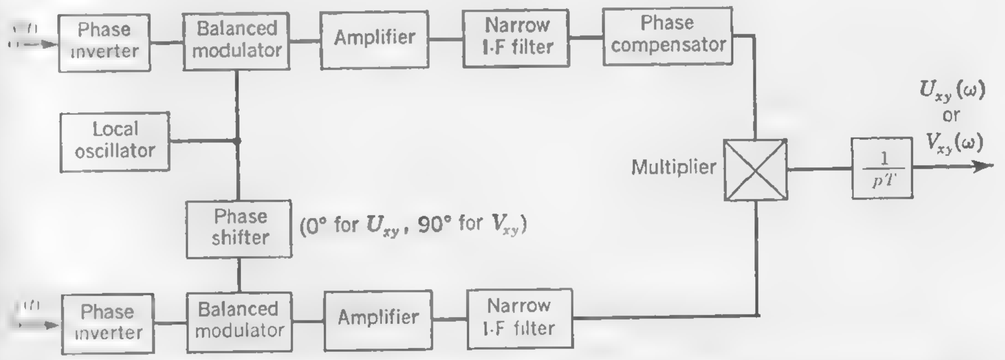


FIG. 5.9.11*b*. This heterodyne-type cross-spectrum analyzer has its 90-deg phase shifter inserted in one of the local oscillator leads when needed. Note the use of a phase compensator to correct for small phase differences in the filters.

If the data-record length is T and the equivalent filter spectral bandwidth is Δf, then the equivalent number of degrees of freedom in the spectral estimate is

$$n = 2T\,\Delta f \tag{5.9.36}$$

or $2/n$ gives the variance of any point in the spectral estimate. For given values of n it is possible to calculate[7,9] "confidence limits" corresponding to various probabilities. Thus if $T = 10$ sec, and $\Delta f = 2$ cps, $n = 40$, and there is 90 per cent probability that the true value of the spectrum at a given frequency is between 71.9 and 151 per cent of the measured value.

5.9.21. Fourier Inversion. The Wiener-Khintchine theorem provides interrelationships between correlation functions and power spectral densities which indicate methods for computing one from another. Thus if $\phi_{xy}(\tau)$ is known, it is possible to determine $\Phi_{xy}(j\omega)$ by using Eq. (33), which may be rewritten as

$$\Phi_{xy}(j\omega) = \int_{-\infty}^{\infty} \phi_{xy}(\tau)\cos\omega\tau\,d\tau - j\int_{-\infty}^{\infty} \phi_{xy}(\tau)\sin\omega\tau\,d\tau \tag{5.9.37}$$

To mechanize this equation, it is only necessary to generate $\phi_{xy}(\tau)$ in a function generator, cos $\omega\tau$ and sin $\omega\tau$ in a single two-integrator plus sign-changer loop, then multiply and integrate for each value of frequency which is of interest (see also Sec. 6.6). Two sources of error will appear and may be dealt with as follows:

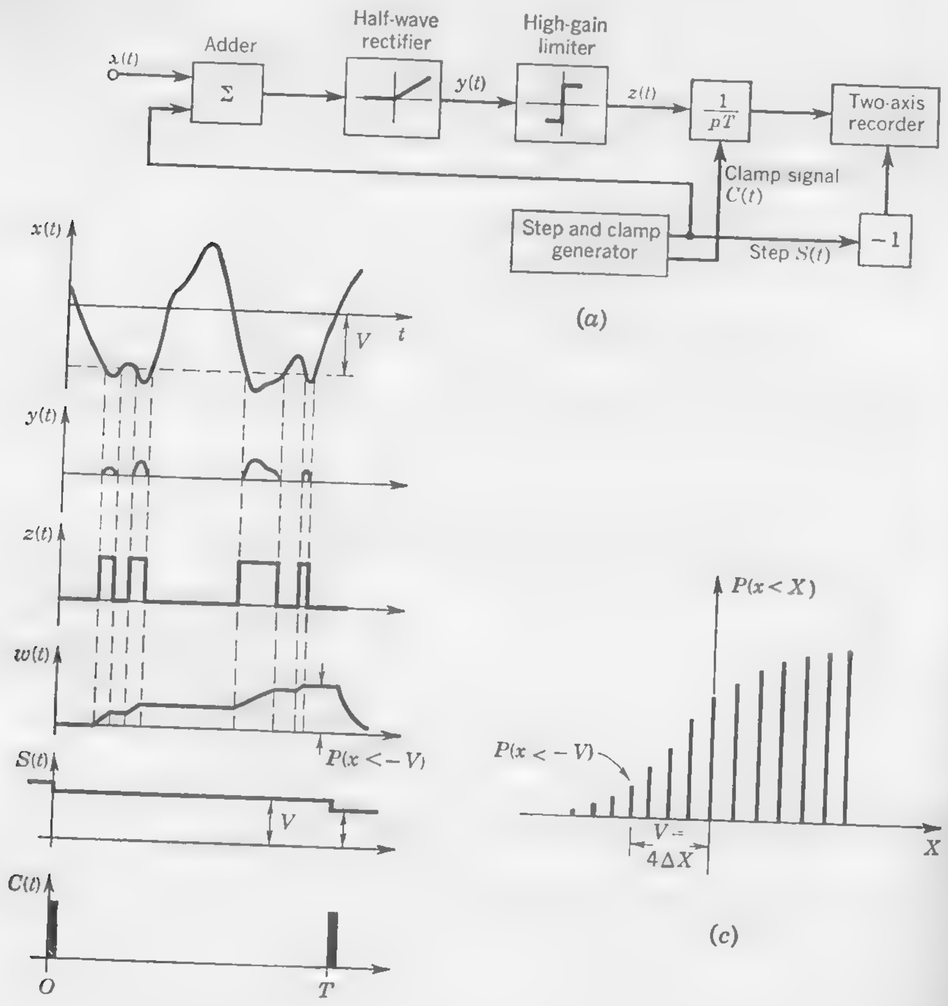


Fig. 5.9.12*a*. Block diagram of a probability-distribution analyzer using repetitive-analog computer equipment.

Fig. 5.9.12*b*. Waveforms in the analyzer. If $2n + 1$ steps in X are required, $S(t)$ must have $2n + 1$ equal steps, each of length T, beginning at $n\,\Delta X$. The clamping pulse must actuate a relay or other device to discharge the integrating capacitor every T sec.

Fig. 5.9.12*c*. Typical output display obtained on a cathode-ray tube or xy recorder.

1. Random errors in the measured estimate of $\phi_{xy}(\tau)$ at the finite steps in τ will yield a function with a random variation superimposed on the true value, and these errors will appear in $\Phi_{xy}(j\omega)$.[25] If the steps in τ are close enough, a smooth curve may be drawn through the measured values of $\phi_{xy}(\tau)$, giving in effect a combination of ensemble averaging and low-pass filtering. This procedure is usually somewhat automatic, for many correlators give 40 to 100 points, and the number of segments available in a diode function generator is ordinarily much less.

2. Truncation errors occur if the correlation function (or spectral density) which is to be transformed is larger than the random error at the largest value of $|\tau|$ (or ω) for which the function has been measured.

The truncation error, which may be particularly troublesome in oceanography and electroencephalography, has been given particular attention for the transformation (14), where it can give rise to false oscillations in the spectral estimate.[23,24] These can be reduced through multiplication of $\phi_{xx}(\tau)$ by well-chosen functions $l(\tau)$ (*lag windows*), which, in effect, round off the truncated time function.[23]

ANALOG COMPUTATION OF PROBABILITY ESTIMATES

5.9.22. Probability-distribution Analyzers. The (cumulative) probability-distribution function $P(X) \equiv P(x < X)$ of a random process must often be measured. A scheme for such measurement, which uses mostly standard analog-computing equipment, is shown in Fig. 5.9.12. This type of measurement, like the direct measurement

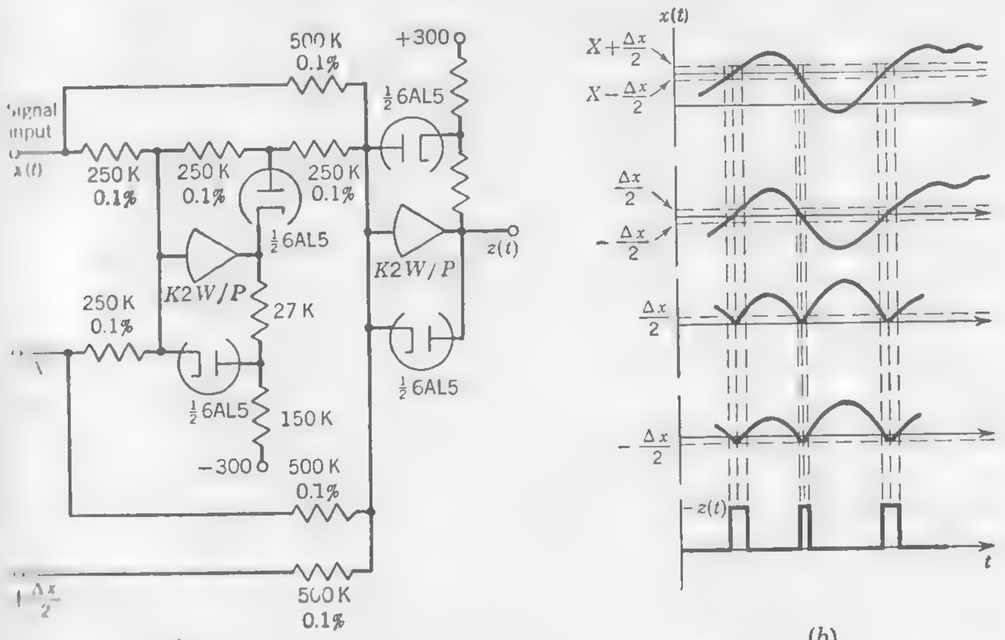


Fig. 5.9.13. Operational-amplifier slicer circuit (*a*) and operations performed (*b*) to yield output pulses of width proportional to the time intervals during which $X - \Delta X/2 < x(t) < X + \Delta X/2$. The slicer circuit combines a precision absolute-value circuit (Table 4.4.1) with a comparator (based on Ref. 27).

of correlation functions, is rather slow unless the equipment shown is duplicated many times to make a parallel machine, or unless "high-speed" or wide-band computer equipment is used, with a corresponding speed-up, if needed, of the input data rate (see also Sec. 6.3).[19]

5.9.23. Probability-density Analyzers. The probability density of a random process $p(X) = dP(X)/dX$ may be obtained by taking the slope of $P(X)$ graphically after it is measured by the method of Fig. 5.9.12.

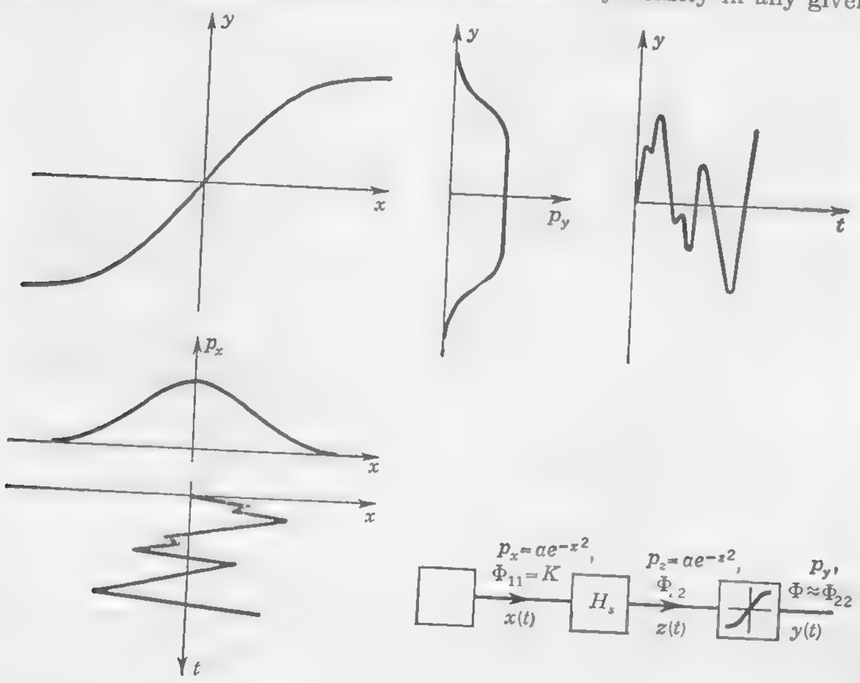
It is also possible to measure the probability density directly. A modification of the circuit shown in Fig. 5.9.12 can produce a train of pulses which are positive (or negative) whenever the random wave is within the amplitude interval $x - \Delta x/2 < x(t) < x + \Delta x/2$. Such a pulse train may be obtained through subtraction of the $x(t)$ waveforms in Fig. 5.9.12 for the two levels $x + \Delta x/2$ and $x - \Delta x/2$,[26] or by the

scheme of Fig. 5.9.13, which uses standard analog computing equipment.[27] The circuit shown may be followed by an integrator and recorder as shown in Fig. 5.9.12, or in either case, a digital read-out scheme may be used. In this case, a pulse train is gated by the pulses $z(t)$ and these pulses are used to actuate a counter.[27]

The probability density determined in this way will itself have random fluctuations because of the finite times of observation. The confidence interval[1] may be determined if the variance σ_p^2 corresponding to a set of measurements p_i of the probability density p can be determined. This can be estimated if several runs are made at the same level, giving as the estimate of the variance,

$$\hat{\sigma}_p^2 = \frac{\sum_{i=1}^{n} (p_i - p_{av})^2}{n - 1}$$

where p_{av} is the average of the n measurements. The estimate of the variance, together with an assumed distribution for the p_i, can be used to find approximate confidence intervals for typical measurements of probability density in any given study.



(*a*) (*b*)

FIG. 5.9.14*a*. Shaping of the probability density by means of a function generator implementing $y = f(x)$. Here the Gaussian distribution p_x of $x(t)$ becomes the nearly rectangular distribution p_y of $y(t)$.
FIG. 5.9.14*b*. Spectrum shaping by linear networks should precede probability density shaping by nonlinear function generators.

5.9.24. Generation of Random Voltages of Given Spectra and Probability Densities. The probability-density function may be molded from the Gaussian (or other) initial density. Thus, as shown in Fig. 5.9.14*a*, a function generator may be used to obtain a nearly rectangular distribution from a Gaussian distribution.

If a certain correlation function or spectral-density function is also prescribed, a linear network designed to give the required characteristics for a "white" noise input should be inserted ahead of the nonlinear function generator used as a probability-distribution shaper, as shown in Fig. 5.9.14*b*. The nonlinear circuit will have some effect on the spectrum (although this will be small in most practical cases), and thus some readjustment of the linear circuit may be necessary.

The nonlinear device need not be an arbitrary-function generator in all cases. Noise of a Rayleigh distribution, for example, may be obtained through linear envelope detection of narrow-band Gaussian noise.[28]

References

1. Cramér, Harald: *Mathematical Methods of Statistics*, Princeton University Press, Princeton, N.J., 1951.
2. Wiener, Norbert: *Extrapolation, Interpolation and Smoothing of Stationary Time Series*, John Wiley & Sons, Inc., New York, 1949.
3. Davenport, W. B., Jr., and W. L. Root: *An Introduction to the Theory of Random Signals and Noise*, McGraw-Hill Book Company, Inc., New York, 1958.
4. Laning, J. H., Jr., and R. H. Battin: *Random Processes in Automatic Control*, McGraw-Hill Book Company, Inc., New York, 1956.
5. Bendat, J. S.: *Principles and Applications of Random Noise Theory*, John Wiley & Sons, New York, 1958.
6. Rice, S. O.: Mathematical Analysis of Random Noise, *Bell System Tech. J.*, **23**:282–332 (July, 1944); **24**:46–157 (January, 1945).
7. Crandall, S. H. (ed.): *Random Vibration*, Technology Press, MIT, Cambridge, Mass., 1958.
8. Mazelsky, B., and H. B. Amey, Jr.: On the Simulation of Random Excitation for Airplane Response Investigations on Analog Computers, *J. Aeronaut. Sci.*, **24**:633–649 (September, 1957).
9. Press, H., and J. W. Tukey: Power Spectral Methods of Analysis and Their Application to Problems in Airplane Dynamics, *AGARD Flight Test Manual*, vol. IV, June, 1956.
10. Haneman, V. S., and J. W. Senders: Correlation Computation on Analog Devices, *J. Assoc. Computing Machinery*, **2**(4):267–279 (October, 1955).
11. Benedict, T. R., H. S. Min, V. C. Rideout, and W. B. Swift: Design of a Delay Line for an Analog Correlator, *Proc. Natl. Electronics Conf.*, **14**:715–723 (1958).
12. Goff, K. W.: An Analog Electronic Correlator for Accoustic Measurements, *J. Acoust. Soc. Am.*, **27**(2):223–236 (March, 1955).
13. Spooner, M. G., and V. C. Rideout: Correlation Studies of Linear and Nonlinear Systems, *Proc. Natl. Electronics Conf.*, **12**:321–335 (1956).
14. Lampard, D. G.: A New Method of Determining Correlation Functions of Stationary Time Series, *Proc. Inst. Elec. Engrs.*, part C, **102**(1):35–41 (March, 1955).
15. Newton, G. C., Jr.: Compensation of Feedback Control Systems Subject to Saturation, *J. Franklin Inst.*, **254**:391–413 (October, November, 1952).
16. Schultz, W. C.: Choice of Measures and Criteria for System Synthesis and Analysis, Ph.D. Thesis, University of Wisconsin, January, 1958.
17. Biernson, G. A.: Fundamental Equations for the Application of Statistical Techniques to Feedback-control Systems, *IRE Trans. on Automatic Control*, PGAC-2, February, 1957, pp. 56–78.
18. Tukey, J. W.: The Sampling Theory of Power Spectrum Estimates, Symposium on Applications of Autocorrelation Analysis, Woods Hole, Mass., 1949, ONR, Washington, D.C.
19. Smith, Francis B.: Analog Equipment for Processing Randomly Fluctuating Data, *Aeronaut. Eng. Rev.*, **14**(5):113–119 (May, 1955).
20. Chang, S. S. L.: An Ocean Wave Power Spectrum Analyzer, *Proc. Natl. Electronics Conf.*, **10**:349–357 (1954).
21. Kishi, F. H., R. N. Boland, and N. Furumoto: Analysis of Low-frequency Noise Using High-pass Filters and Analog Computer Techniques, *Proc. Natl. Electronics Conf.*, **11**:212–221 (1955).
22. Uberoi, M. S., and E. G. Gilbert: Technique for Measurement of Cross-spectral Density of Two Random Functions, *Rev. Sci. Instr.*, **30** (March, 1959).
23. Blackman, R. B., and J. W. Tukey: The Measurement of Power Spectra from the Point of View of Communication Engineering, *Bell System Tech. J.*, **37**(1):185–282 (January, 1958); (2):485–569 (March, 1958).
24. Ross, Douglas T.: Improved Computational Techniques for Fourier Transformation, MIT Servomechanisms Lab., *Rept.* 7138-R-5, June 25, 1954.
25. Alper, Paul: *Propagation of Random Errors in Correlation Functions into the Power Spectra*, M. S. Thesis, Univ. of Wisconsin (1960).

26. Davis, H. E., and G. R. Cooper: Direct-reading Probability Distribution Meter, *Proc. Natl. Electronics Conf.*, **10**:358–366 (1954).
27. Caldwell, W. F., G. A. Korn, V. R. Latorre, and G. R. Peterson: A Precision Amplitude-distribution Analyzer, *Trans. IRE PGEC*, June, 1960.
28. Favreau, R. R., H. Low, and I. Pfeffer: Evaluation of Complex Statistical Functions by an Analog Computer, National Convention Record of the IRE, 1956.
29. Wang, M. C., and G. E. Uhlenbeck: On the Theory of the Brownian Motion II, *Revs. Modern Phys.*, **17**(2, 3):323–342 (April–July, 1945).
30. Korn, G. A., and T. M. Korn: *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill Book Company, Inc., New York, 1961.

PART 10: SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

By R. M. Howe

INTRODUCTION

5.10.1. Requirement for the Solution of Partial Differential Equations. Since the electronic analog computer can integrate with respect to only one variable, namely, time, it is necessary to convert a partial differential equation to one or more ordinary differential equations in order to solve it on the analog computer, either (1) by the method of separation of variables, leading to an eigenvalue problem; or (2) by the finite-difference method wherein derivatives with respect to all variables but one are replaced by finite-difference approximations.

5.10.2. Example Types of Partial Differential Equations. In order to illustrate the different analog-computer techniques for solving partial differential equations several example equations will be treated, including the following:

$$k\frac{\partial \phi}{\partial t} = \nabla^2\phi + f \qquad \text{(HEAT EQUATION, OR DIFFUSION EQUATION)} \qquad (5.10.1)$$

$$k\frac{\partial^2 \phi}{\partial t^2} = \nabla^2\phi + f \qquad \text{(WAVE EQUATION)} \qquad (5.10.2)$$

$$k\frac{\partial^2 \phi}{\partial t^2} = \nabla^4\phi + f \qquad \text{(STRUCTURAL-BEAM EQUATION)} \qquad (5.10.3)$$

where ϕ is the dependent variable, t is the independent variable time, f is the forcing function, and ∇^2 is the Laplacian operator. Thus, in rectangular coordinates $\nabla^2\phi = \partial^2\phi/\partial x^2$ if there is one distance variable x, $\nabla^2\phi = \partial^2\phi/\partial x^2 + \partial^2\phi/\partial y^2$ if there are two distance variables x and y, etc. Similarly, in one dimension $\nabla^4\phi = \partial^4\phi/\partial x^4$, in two dimensions $\nabla^4\phi = \partial^4\phi/\partial x^4 + 2\partial^4\phi/\partial x^2\,\partial y^2 + \partial^4\phi/\partial y^4$.

Equations (1) to (3) are shown in their most simple form. In treating detailed examples, we shall include the possibility of variable medium characteristics, nonlinearities, coordinate systems other than rectangular, etc.

5.10.3. Importance of Error Analysis. It is important to realize that the techniques of converting the partial differential equation to ordinary differential equations lead only to an approximate mathematical representation of the original equation. For this reason it is vital to have some measure of the error which the approximation produces, particularly since in many cases the error can be reduced only by adding more analog-computing equipment.

SOLUTION BY SEPARATION OF VARIABLES

5.10.4. Basic Equation for Heat Flow. Consider the temperature $u(\bar{x},\bar{t})$ between the two infinite slabs in Fig. 5.10.1. Here $\bar{x}$ is distance through the slab and $\bar{t}$ is the time variable. The left boundary occurs at $\bar{x} = 0$ and the right boundary at $\bar{x} = L$.

The equation of heating-rate equilibrium is

$$C(\bar{x})\frac{\partial u(\bar{x},\bar{t})}{\partial \bar{t}} = \frac{\partial}{\partial \bar{x}}\left[K(\bar{x})\frac{\partial u(\bar{x},\bar{t})}{\partial \bar{x}}\right] + \bar{S}(\bar{x},\bar{t}) \qquad (5.10.4)$$

where K is the thermal conductivity and C is the heat-capacity constant for the medium. In general K and C are functions of the distance variable $\bar{x}$. $\bar{S}$ represents the rate of heat supplied per unit volume by sources within the medium. The heat flux $F_{\bar{x}}$ along the $\bar{x}$ direction is given by

$$F_{\bar{x}} = -K\frac{\partial u}{\partial \bar{x}} \qquad (5.10.5)$$

Let us introduce dimensionless variables to make Eq. (4) easier to scale later on. First define a dimensionless distance variable x given by

$$x = \frac{\bar{x}}{L} \qquad (5.10.6)$$

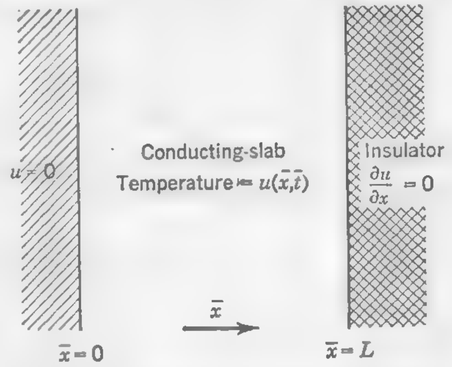


FIG. 5.10.1. Transient heat-flow problem in a slab.

This makes the thickness of the slab unity in x. Next let us transform the variable characteristics of heat capacity $C(\bar{x})$ and conductivity $K(\bar{x})$ into dimensionless variables $\phi_c(x)$ and $\phi_k(x)$, respectively. Thus, let

$$C(Lx) = C_0\phi_c(x) \qquad K(Lx) = K_0\phi_k(x) \qquad (5.10.7)$$

Here C_0 and K_0 are constants equal to the maximum value of $C(\bar{x})$ and $K(\bar{x})$, respectively. For a uniform medium $\phi_c(x) = \phi_k(x) = 1$. From Eqs. (6) and (7), Eq. (4) becomes

$$\frac{C_0L^2}{K_0}\phi_c(x)\frac{\partial u}{\partial \bar{t}} = \frac{\partial}{\partial x}\phi_k(x)\frac{\partial u}{\partial x} + \frac{L^2}{K_0}\bar{S}(Lx,\bar{t}) \qquad (5.10.8)$$

We introduce a dimensionless time variable t given by

$$t = \frac{K_0}{C_0L^2}\bar{t} \qquad (5.10.9)$$

from which Eq. (8) becomes

$$\phi_c(x)\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\phi_k(x)\frac{\partial u}{\partial x} + S(x,t) \qquad (5.10.10)$$

where

$$S(x,t) = \frac{L^2}{K_0}\bar{S}\left(Lx, \frac{C_0L^2}{K_0}t\right) \qquad (5.10.11)$$

For the special case of a uniform medium with no internal heat sources Eq. (10) becomes

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \qquad (5.10.12)$$

which is essentially the same as Eq. (1) for the one-dimensional case.

For the example in Fig. 5.10.1 the left boundary is held at zero temperature and the right boundary is insulated (no heat flux). Thus

$$u(0,t) = 0 \qquad (5.10.13)$$

$$\frac{\partial u}{\partial x}(1,t) = 0 \qquad (5.10.14)$$

Let us denote the initial temperature distribution by $U(x)$, i.e.,

$$u(x,0) = U(x) \tag{5.10.15}$$

5.10.5. Separation of Variables. Consider the case where $S(x,t) = 0$ (no heat sources), and assume that

$$u(x,t) = X(x)T(t) \tag{5.10.16}$$

Substituting Eq. (16) into Eq. (10) and dividing through by $\phi_c XT$, we have

$$\frac{1}{T(t)}\frac{dT(t)}{dt} = \frac{1}{\phi_c(x)X(x)}\frac{d}{dx}\phi_k(x)\frac{dX(x)}{dx} \tag{5.10.17}$$

Since the left side of Eq. (17) is a function only of t and the right side is a function only of x, they can be equal for all t and x only if both are equal to the same constant. Hence

$$\frac{1}{T(t)}\frac{dT(t)}{dt} = -\beta = \text{const}$$

or

$$\frac{dT}{dt} + \beta T = 0 \tag{5.10.18}$$

In the same way

$$\frac{d}{dx}\phi_k(x)\frac{dX}{dx} + \beta\phi_c(x)X = 0 \tag{5.10.19}$$

The solution of Eq. (18) is simply an exponential function with decay constant β, i.e.,

$$T = Ae^{-\beta t} \tag{5.10.20}$$

where A is an arbitrary constant. Equation (19) is subject to the boundary conditions of the original problem, since homogeneous boundary conditions on $u(x,t)$ must also be satisfied by $X(x)$:

$$X(0) = \frac{dX}{dx}(1) = 0 \tag{5.10.21}$$

The solutions to Eq. (19) can satisfy the boundary conditions of Eq. (21) only for discrete values β_n of the constant β. The solution $X_1(x)$ corresponding to the smallest allowable value β_1 gives the temperature distribution through the medium corresponding to the lowest mode of exponential decay with time. $X_2(x)$ gives the shape of the second mode, which decays exponentially with decay constant β_2, etc. The discrete values β_n are known as *eigenvalues*, and the corresponding solutions $X_n(x)$ are known as *eigenfunctions* or *normal modes*. The higher the mode (i.e., the larger the integer n), the larger the decay constant β_n, and the faster will that particular mode shape $X_n(x)$ decay or die out.

When using the method of separation of variables on the heat equation, the time-dependent equation always takes the form of Eq. (18) and hence the solution is always the exponential function of Eq. (20). The exact form of Eq. (19) for the spatial part of the solution depends on the coefficients $\phi_k(x)$ and $\phi_c(x)$ which give the variation of conductivity and heat capacity, respectively, through the medium. This equation, subject to the boundary conditions of Eq. (21), can be solved usefully with the electronic analog computer, with computer time corresponding to the dimensionless distance variable x. Let us first consider the general form of the solution.

The heat equation (10) is satisfied by solutions of the form $A_1X_1(x)e^{-\beta_1 t}$, $A_2X_2(x)e^{-\beta_2 t}$, etc. The superposition principle applies, and every linear combination of the above solutions will be a solution

$$u(x,t) = \sum_{n=1}^{\infty} A_nX_n(x)e^{-\beta_n t} \tag{5.10.22}$$

The constants A_n can be evaluated by use of the orthogonality properties of $X_n(x)$. From Eq. (10) and the boundary conditions of Eq. (21) it is easy to show that[1]

$$\int_0^1 \phi_c(x)X_n(x)X_m(x)\,dx = \begin{cases} 0 & n \neq m \\ N_n & n = m \end{cases} \tag{5.10.23}$$

where n and m are integers; N_n is called the normalizing constant. From Eqs. (15) and (22) we have at $t = 0$

$$U(x) = \sum_{n=1}^{\infty} A_nX_n(x) \tag{5.10.24}$$

Multiplying both sides by $\phi_c(x)X_m(x)$ and integrating with respect to x from 0 to 1 it follows from Eq. (23) that

$$A_n = \frac{1}{N_n}\int_0^1 \phi_c(x)U(x)X_n(x)\,dx \tag{5.10.25}$$

where

$$N_n = \int_0^1 \phi_c(x)X_n^2(x)\,dx \tag{5.10.26}$$

The analog computer can be used not only to determine the normal-mode functions $X_n(x)$ and eigenvalues β_n but also, if desired, to solve for the coefficients A_n necessary to have $u(x,t)$ satisfy the prescribed initial temperature distribution across the slab.

$\frac{d}{dx}\phi_k(x)\frac{dX}{dx} + \beta\phi_c(x)X = 0;\ X(0) = \frac{dX}{dx}(1) = 0$

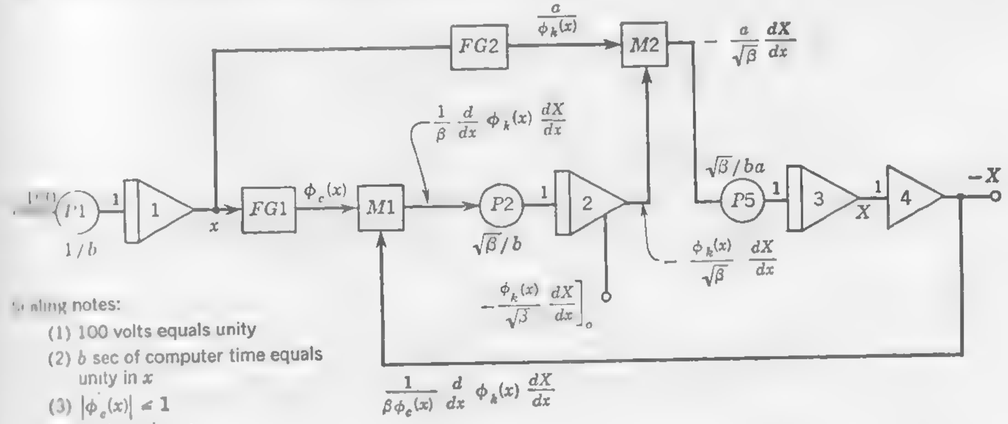


FIG. 5.10.2. Computer circuit for solving eigenvalue problem.

5.10.6. Computer Circuit for the Boundary-value Problem. The computer circuit for solving Eq. (19) is shown in Fig. 5.10.2. Time on the computer represents distance x through the slab. Although Eq. (19) is actually linear (the coefficients are known functions of the independent variable x), nonlinear equipment in the form of two function generators and two multipliers is needed to mechanize the equations. The boundary conditions of Eq. (21) imply that at $x = 0$ the output of amplifier 3 equals zero, while at $x = 1$ (b sec after the solution begins) the output of amplifier 2 equals zero. This latter condition will be satisfied, of course, only if a correct eigenvalue β_n is set into potentiometers $P2$ and $P3$. Thus a series of trial-and-error solutions must be generated. The computer solution length b should be set at some convenient value, say 10 sec. An arbitrary initial condition is set on amplifier 2, representing the slope dX/dx at $x = 0$. Then a trial value for β is set on $P2$ and $P3$, and a solution is run. The output $-(\phi_k/\sqrt{\beta})\,dX/dx$ of amplifier 2 is recorded and should go

through zero b sec later. It will not, of course, and the solution must be repeated with a new trial β value. After several such runs it is easy to converge on the correct solution, as shown in Fig. 5.10.3. An accurate xy plotter is convenient for recording purposes; if a conventional strip-chart recorder is used the record should be "blown up" near the end of the solution in order to have a more accurate indication of when the final boundary condition is met. Since this is a linear problem, the shape of the

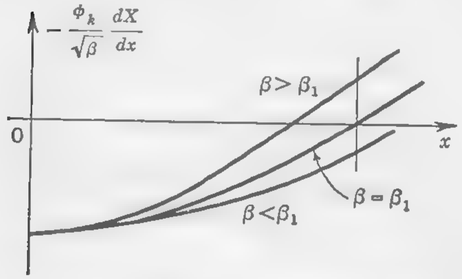


FIG. 5.10.3. Trial-and-error convergence on first-mode solution.

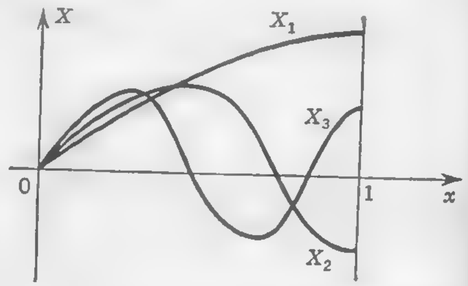


FIG. 5.10.4. Solution for first three normal modes.

solution is independent of the amplitude. Thus once a near-correct eigenvalue is obtained, the initial condition on amplifier 2 should be increased until solutions are generated which utilize near full-scale voltage range on the outputs of amplifiers 2 and 3. By distributing the gain equally between potentiometers 2 and 3 the maximum voltage excursion at the outputs of amplifiers 2 and 3 should be comparable. When an exact, full-amplitude solution is obtained, the value of β is recorded. Then the solution is rerun, recording X vs. x.

Equations being solved: $A_nN_n = \int_0^1 \phi_c(x)U(x)X_n dx$; $N_n = \int_0^1 \phi_c(x)X_n^2 dx$

Note: Scaling same as Fig. 5.10.2
x, $\phi_c(x)$, $X_n^2(x)$ computed in Fig. 5.10.2

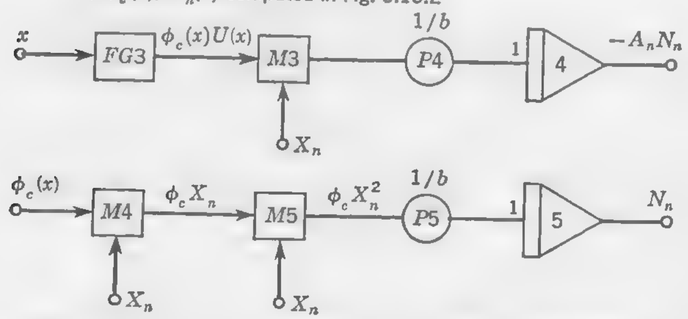


FIG. 5.10.5. Circuit for generating Fourier coefficients.

For a uniform medium $\phi_c = \phi_k = 1$ and Eq. (19) represents simply a mass-spring system with a period of $2\pi b/\sqrt{\beta}$ sec. For the first mode the solution of Fig. 5.10.3 represents one-quarter of a period, so that the total period would be $4b$ sec. Thus $4b = 2\pi b/\sqrt{\beta_1}$, and $\beta_1 = (\pi/2)^2$. The second mode represents three-quarters of a period, the third five-quarters, etc., so that $\beta_2 = (3\pi/2)^2$, $\beta_3 = (5\pi/2)^2$, etc. If ϕ_c and ϕ_k do not vary appreciably from unity in the case of a nonuniform medium, the above values of β can be used as first estimates for a point of departure in the trial-and-error process. Typical solutions for the first three modes are shown in Fig. 5.10.4. It is usually best to determine the lowest mode first, the second mode next, etc.

5.10.7. Generation of the Fourier Coefficients. Frequently the normal-mode functions X_n and eigenvalues β_n are the only quantities required from the computer.

For example, in a structural dynamics problem solved by separation of variables the eigenvalues represent the normal-mode frequencies and the eigenfunctions or normal-mode shapes. But to solve the complete problem with specified initial conditions we need to find the coefficients A_n given in Eq. (25). This can be accomplished simultaneously with the determination of the eigenfunctions X_n in the previous section. The circuit is shown in Fig. 5.10.5. Note that the normalizing constant N_n must be computed in order to calculate A_n from $X_n(x)$ and $U(x)$. Because a series of orthogonal functions as given in Eq. (24) is called a Fourier series, the constants A_n are called Fourier coefficients.

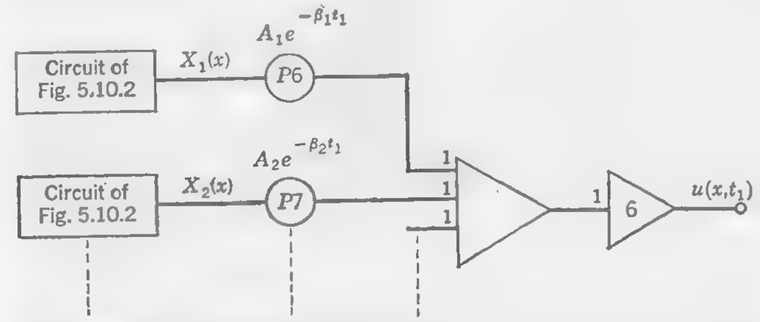


FIG. 5.10.6. Circuit for generating temperature distribution across the slab at $t = t_1$.

5.10.8. Analog Computation of the Complete Series Solution. The complete solution to the transient-heat-flow problem of Fig. 5.10.1 for the particular initial temperature distribution $U(x)$ is given by the infinite series of Eq. (22). Each eigenfunction $X_n(x)$ is multiplied by a decreasing exponential $e^{-\beta_n t}$ and has a Fourier-coefficient weighting factor A_n. In practice, of course, we need only calculate a finite number of terms, hopefully only the first several. Since the higher modes have large β_n and decay very rapidly, it is possible to neglect these terms after the first small time interval has elapsed. Then too, the coefficients A_n decrease in size for large n. After the normal modes X_n, eigenvalues β_n, and Fourier coefficients A_n have been calculated with the analog computer as described in the previous two sections, the separate terms in Eq. (22) can be generated simultaneously on the computer and added together with weighting factors $A_ne^{-\beta_n t_1}$ to represent the temperature distribution across the slab at time t_1. Although this requires that the circuit of Fig. 5.10.2 be duplicated N times, where N is the number of terms considered, it is a relatively simple circuit, and the functions $\phi_c(x)$ and $\phi_k(x)$ need only be generated once. The over-all circuit for accomplishing this is shown in Fig. 5.10.6. For any time t_1, potentiometers $P6$, $P7$, etc., are simply set to $A_1e^{-\beta_1 t_1}$, $A_2e^{-\beta_2 t_1}$, etc. The different X_n solutions are then run simultaneously and the output of amplifier 6 can be recorded to yield a plot of temperature distribution across the slab at t_1. The runs can be repeated for t_2, t_3, etc. In Fig. 5.10.7 is shown a set of typical results obtained in this manner for a uniform slab with a uniform initial temperature distribution. Note that for $t > 0.16$ the distribution is essentially sinusoidal.

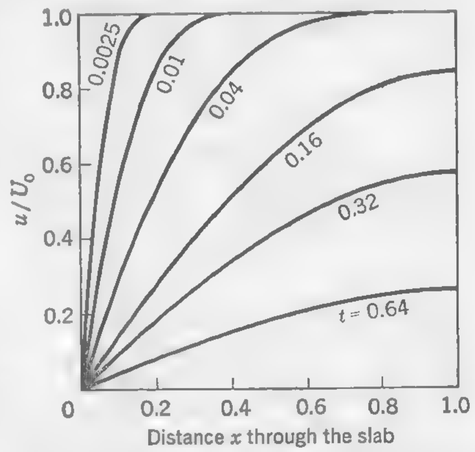


FIG. 5.10.7. Transient temperature distribution across a uniform slab.

If the complete solution is desired, the method of separation of variables described here should be weighed carefully in comparison with the difference-equation technique of Art. 5.10.12., which gives a much more direct result.

5.10.9. Additional Eigenvalue Problems. (*a*) *Beam Equation.* In its simplest form the equation for the transverse displacement y of a uniform structural beam is given by Eq. (3). For a nonuniform beam with one spatial coordinate x the equation can be written

$$\frac{\partial^2}{\partial x^2}\phi_f(x)\frac{\partial^2 y}{\partial x^2} + \phi_d(x)\frac{\partial^2 y}{\partial t^2} = f(x,t) \tag{5.10.27}$$

where x = dimensionless distance along the beam (the beam length is unity in x)
t = dimensionless time
y = transverse displacement, a function of x and t
$\phi_f(x)$ = dimensionless stiffness variation along the beam
$\phi_d(x)$ = dimensionless mass distribution along the beam
$f(x,t)$ = external force applied along the beam.

The technique for obtaining Eq. (27) from the original equation in dimensional form is analogous to the method used to obtain Eq. (10) from Eq. (4) in the case of heat flow.[2] Boundary conditions depend on the type of end fastening[3] and are summarized below:

Type of End Fastening	*Boundary Condition*
Hinged (simply supported)	$y = 0,\ \partial^2 y/\partial x^2 = 0$
Built-in	$y = 0,\ \partial y/\partial x = 0$
Free	$\partial^2 y/\partial x^2 = 0,\ \partial/\partial x(\phi_f \partial^2 y/\partial x^2) = 0$

Note that $\phi_f \partial^2 y/\partial x^2$ is proportional to the bending moment and $\partial/\partial x(\phi_f \partial^2 y/\partial x^2)$ is proportional to the shear force. Assume $f(x,t) = 0$. To separate variables let $y(x,t) = X(x)T(t)$ and substitute into Eq. (27). The resulting equation for T is

$$\frac{d^2T}{dt^2} + \beta^2 T = 0 \tag{5.10.28}$$

where β is a constant. The solution for $\beta > 0$ is simply $T(t) = A\cos\beta t + B\sin\beta t$ where A and B are arbitrary constants. The equation in X is

$$\frac{d^2}{dx^2}\phi_f(x)\frac{d^2X}{dx^2} - \beta^2\phi_d(x)X = 0 \tag{5.10.29}$$

This equation must satisfy the boundary conditions, which in turn depend on the type of end condition. For example, if we have a cantilever beam with the end at $x = 0$ built in, the boundary conditions become

$$X(0) = X'(0) = 0 \qquad X''(1) = [\phi_f X'']'(1) = 0 \tag{5.10.30}$$

Equations (29) and (30) represent an eigenvalue problem, since only for discrete values of β can the boundary conditions be satisfied. To solve Eq. (29) with an analog computer we let time on the computer represent distance x along the beam. The circuit will be similar to the schematic in Fig. 5.10.2, with five integrators required instead of three. A trial value of β is chosen and zero initial conditions are used for the integrators with outputs X and X'. Trial initial conditions are set on $\phi_f X''$ and $[\phi_f X'']'$. Since the problem is linear, only the ratio $[\phi_f X'']'(0)/\phi_f X''(0)$ changes the nature of the solution. Both $\phi_f X''$ and $[\phi_f X'']'$ are recorded, and the ratio of initial conditions is varied until a solution is obtained for which $\phi_f X''(1) = 0$, i.e., a solution for which the voltage representing $\phi_f X''$ goes through zero b sec after the solution begins, where b sec on the computer represents unity in x. For a correct solution $[\phi_f x'']'$ should also vanish at $x = 1$. It will not, of course, since the correct value of β will not have been chosen for the first trial. If $[\phi_f X'']'$ goes through zero for $x < 1$, a smaller value of β is needed for the next trial. If $[\phi_f X'']' = 0$ for $x > 1$, a larger value of β is needed. Actually, this can be observed by examining the $\phi_f X''$ record

alone, since $[\phi_f X'']'$ is simply the slope of $\phi_f X''$. This is apparent in Fig. 5.10.8*a*. The initial condition on $\phi_f X''$ should be set so that full-scale voltage excursions at integrator outputs are obtained.

The above procedure leads to a first-mode solution. Higher modes are obtained in the same way by starting with a larger value of β. Second- and third-mode solutions are shown in Fig. 5.10.8*b*. Since Eq. (29) represents inherently an unstable system, it becomes difficult to obtain the higher modes. This is best illustrated for the case of a uniform beam, i.e., for $\phi_f = \phi_d = 1$. The actual solution then becomes

$$X = C_1\cos\beta x + C_2\sin\beta x + C_3 e^{-\beta x} + C_4 e^{\beta x} \tag{5.10.31}$$

where C_1, C_2, C_3, and C_4 are constants and where x is time on the computer. For large β (i.e., for higher modes) the term $C_4 e^{\beta x}$ eventually overwhelms the others. Also, small changes in initial conditions affect the solution greatly. Because of dielectric absorption in the integrating capacitors, random fluctuations in function generators and multipliers, etc., successive solutions may not repeat exactly, making convergence to a correct solution difficult for modes higher than the third. For symmetrical beams this situation can be improved by looking for symmetrical and antisymmetrical modes separately. This allows consideration of only one-half the beam. Thus a free-free symmetrical beam has the following boundary conditions for symmetrical vibration: $\phi_f X''(0) = [\phi_f X''(0)]' = 0$, $X'(\frac{1}{2}) = [\phi_f X''(\frac{1}{2})]' = 0$; for antisymmetrical vibration the conditions become: $\phi_f X''(0) = [\phi_f X''(0)]' = 0$, $X(\frac{1}{2}) = \phi_f X''(\frac{1}{2}) = 0$. By requiring a solution only from $x = 0$ to $x = \frac{1}{2}$, the effect of instability in the solution is greatly reduced.

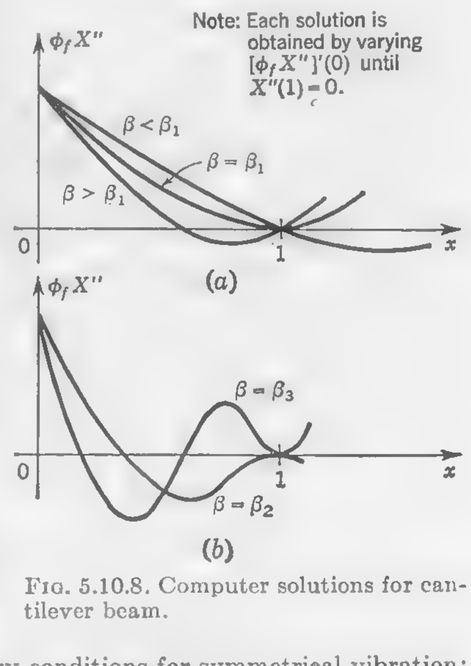


FIG. 5.10.8. Computer solutions for cantilever beam.

The normal-mode solution $X_n(x)$ as obtained above represents the shape of the nth mode of vibration of the beam when oscillating freely at the frequency β_n.

(*b*) *Coupled Bending and Torsion.* Sometimes it is necessary to include twisting as well as bending in a beam, e.g., the problem of an aircraft wing. Then the partial differential equations for a uniform beam take the form

$$\begin{aligned} \frac{\partial^4 y}{\partial x^4} + \frac{\partial^2 y}{\partial t^2} + a_1\theta + a_2\frac{\partial^2\theta}{\partial t^2} &= f_1(x,t) \\ \frac{\partial^2\theta}{\partial x^2} + a_3\frac{\partial^2\theta}{\partial t^2} + a_4 y + a_5\frac{\partial^2 y}{\partial t^2} &= f_2(x,t) \end{aligned} \tag{5.10.32}$$

Here y is the transverse displacement of the beam and θ is the angle of twist. Both elastic and inertial coupling terms are present. For $f_1 = f_2 = 0$ we can separate variables, i.e., assume $y(x,t) = X(x)T(t)$, $\theta(x,t) = \Theta(x)T(t)$. The resulting set of equations in x is the following:

$$\begin{aligned} \frac{d^4X}{dx^4} - \beta^2 X + (a_1 - a_2\beta^2)\Theta &= 0 \\ \frac{d^2\Theta}{dx^2} - a_3\beta^2\Theta + (a_4 - a_5\beta^2)X &= 0 \end{aligned} \tag{5.10.33}$$

For a cantilever beam the boundary conditions of Eq. (30) apply and, in addition, $\Theta(0) = 0$, $\Theta'(1) = 0$.

Here we have a sixth-order eigenvalue problem which can be solved by letting time on the computer be distance x along the beam. Three initial conditions are unknown and must be guessed, namely, $\phi_f X''(0)$, $[\phi_f X'']'(0)$, and $\Theta'(0)$. The ratio of any two to the third is varied until the final conditions are satisfied, namely,

$$\phi_f X''(1) = [\phi_f X'']'(1) = \Theta'(1) = 0$$

This can occur only if β is set at a correct eigenvalue β_n. Thus a three-dimensional convergence problem is present. However, since the beam is uniform, there are no variable coefficients in Eq. (33). This means that the length b of the computer solution can be left arbitrary, so that the end condition becomes the simultaneous vanishing of $\phi_f X''$, $[\phi_f X'']'$, and Θ'. The computer time b at which this occurs defines the relationship between computer time and x, and the three-dimensional convergence problem is reduced to a two-dimensional one. For a nonuniform beam this simplification is not possible.

5.10.10. Use of Normal Coordinates to Represent a Continuous System. In finding the normal-mode shapes $X_n(x)$ for the beam in the previous section we solved the homogeneous equation, i.e., the case where the external force $f(x,t)$ along the beam is zero. For actual applied force $f(x,t)$ it turns out that the beam displacement $y(x,t)$ can be represented in terms of the normal-mode solutions. Thus

$$y(x,t) = \sum_{n=1}^{\infty} X_n(x) q_n(t) \tag{5.10.34}$$

where $q_n(t)$ are the normal coordinates and satisfy the differential equation

$$M_n \ddot{q}_n + M_n \beta_n^2 q_n = F_n(t) \tag{5.10.35}$$

Here M_n and F_n are the generalized mass and force of the nth mode, respectively, and are given by

$$M_n = \int_0^1 \phi_d(x) X_n^2(x)\, dx \tag{5.10.36}$$

and

$$F_n(t) = \int_0^1 f(x,t) X_n(x)\, dx \tag{5.10.37}$$

The generalized mass M_n is actually the same as the normalizing factor N_n given earlier in Eq. (26) and can be computed with the circuit shown in Fig. 5.10.5, where distance x along the beam is time on the computer.

The generalized force F_n is particularly simple to calculate if $f(x,t)$ is concentrated at one point x_0 along the beam (for example, the reaction force on a ballistic missile due to gimbaled engine, where the missile is considered a free-free beam). If $f_0(t)$ represents the concentrated force, then $f(x,t) = \delta(x - x_0) f_0(t)$, where $\delta(x - x_0)$ is a unit impulse along x occurring at $x = x_0$. From Eq. (37) $F_n(t)$ becomes

$$F_n(t) = X_n(x_0) f_0(t) \tag{5.10.38}$$

If $f(x,t)$ is distributed along x, it can always be approximated by a finite number of concentrated forces, and $F_n(t)$ becomes a linear combination of these forces.

The method of simulating a forced beam with the electronic analog computer is now evident. Equation (35) is set up for each normal coordinate q_n with a two-integrator one-summer circuit, where the normal-mode frequency β_n and mode shape $X_n(x)$ (and hence M_n and F_n) have been determined previously with the computer. Then the normal coordinates q_n are added in a summing amplifier in accordance with Eq. (34) to obtain the beam displacement $y(x,t)$ at some particular point x, along the beam, e.g., the point at which a control-system sensing element might be located. In practice only the first several modes need to be considered. Damping can easily be introduced into Eq. (35). External forces $f(x,t)$ which depend on the coordinate q_n, as in the aeroelastic case, can also be included. Thus the normal coordinates give a method of representing a continuous structure by means of a finite number of degrees of freedom with an accuracy that is considerably better than the difference-equation method for the same number of degrees of freedom. The normal-mode shapes and frequencies do need to be calculated ahead of time, however. Sometimes this calculation is done digitally. The same normal-coordinate scheme can of course be applied to the heat equation for the case where external heat sources are present.

5.10.11. Appraisal of the Method of Separation of Variables. This technique is applicable only to linear partial differential equations and can be simply applied only when the boundary conditions are homogeneous. In the case of arbitrary initial conditions it is possible to obtain complete solutions with the analog computer by superposing the normal-mode solutions. Figure 5.10.6 shows the circuit in the case of one-dimensional heat-flow problems. For more complicated problems the determination of the complete solution using this method is probably impractical. However, the determination of the individual mode solutions (eigenfunctions) and eigenvalues can be quite useful in leading to a simple finite-degree-of-freedom representation of the continuous system by means of normal coordinates as discussed in the previous article. Accuracy of normal-mode solutions will be the same order as individual component accuracies. However, many eigenvalue problems have inherent instabilities which make it difficult to obtain the solutions with the analog computer, particularly for high modes. Interestingly enough, these instabilities are seldom in the original problem but are introduced when the variables are separated [see, for example, Eq. (31) in the section on the vibration of beams].

SOLUTION BY DIFFERENCE TECHNIQUES

5.10.12. One-dimensional Heat Flow. Consider the heat-flow problem shown in Fig. 5.10.1 and given in Eq. (10). Instead of computing the temperature u at all x

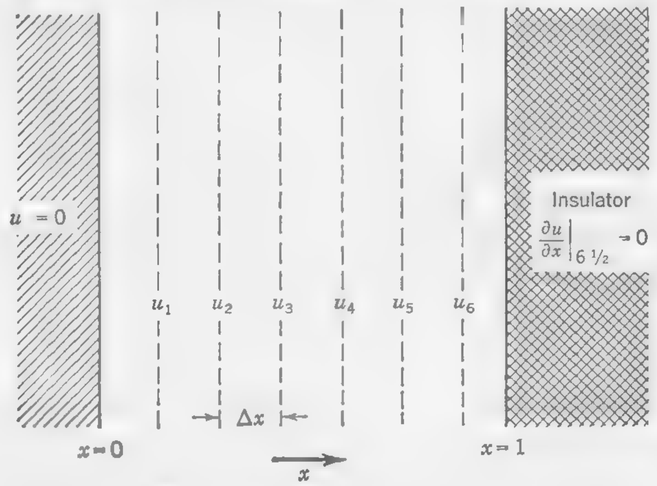


FIG. 5.10.9. Station arrangement for heat flow in a slab.

values, let us compute u only at certain stations along x, as shown in Fig. 5.10.9. Let u_1 be the value of u at the first x station, u_2 at the second, u_n at the nth station. Further, let the distance between stations be a constant Δx. Thus $u(x,t)$ is replaced by $u_1(t)$, $u_2(t)$, . . . , etc., and we can approximate $\partial u/\partial x$ at the $n - \frac{1}{2}$ station by

$$\left.\frac{\partial u}{\partial x}\right|_{n-\frac{1}{2}} \cong \frac{u_n - u_{n-1}}{\Delta x} \tag{5.10.39}$$

This is known as a first-order central-difference approximation. Later we shall consider other approximation formulas in addition to analyzing the error in the approxi-

mation. In a manner analogous to Eq. (39) we can approximate $(\partial/\partial x)\phi_k(\partial u/\partial x)$ at the nth station by

$$\frac{\partial}{\partial x}\phi_k\frac{\partial u}{\partial x}\bigg|_n \cong \frac{1}{\Delta x}\left[\phi_k\frac{\partial u}{\partial x}\bigg|_{n+\frac{1}{2}} - \phi_k\frac{\partial u}{\partial x}\bigg|_{n-\frac{1}{2}}\right] \tag{5.10.40}$$

From Eq. (10) and using Eqs. (39) and (40) we have for the equation of equilibrium at the nth station

$$\phi_{c_n}\frac{du_n}{dt} = \frac{1}{(\Delta x)^2}\left[\phi_{k_{n+\frac{1}{2}}}(u_{n+1} - u_n) - \phi_{k_{n-\frac{1}{2}}}(u_n - u_{n-1})\right] + S_n(t) \tag{5.10.41}$$

where $S_n(t)$ is proportional to the heat flow per unit volume coming from the heat source at the nth station. Note in Eq. (41) that the time derivative of u_n is a total

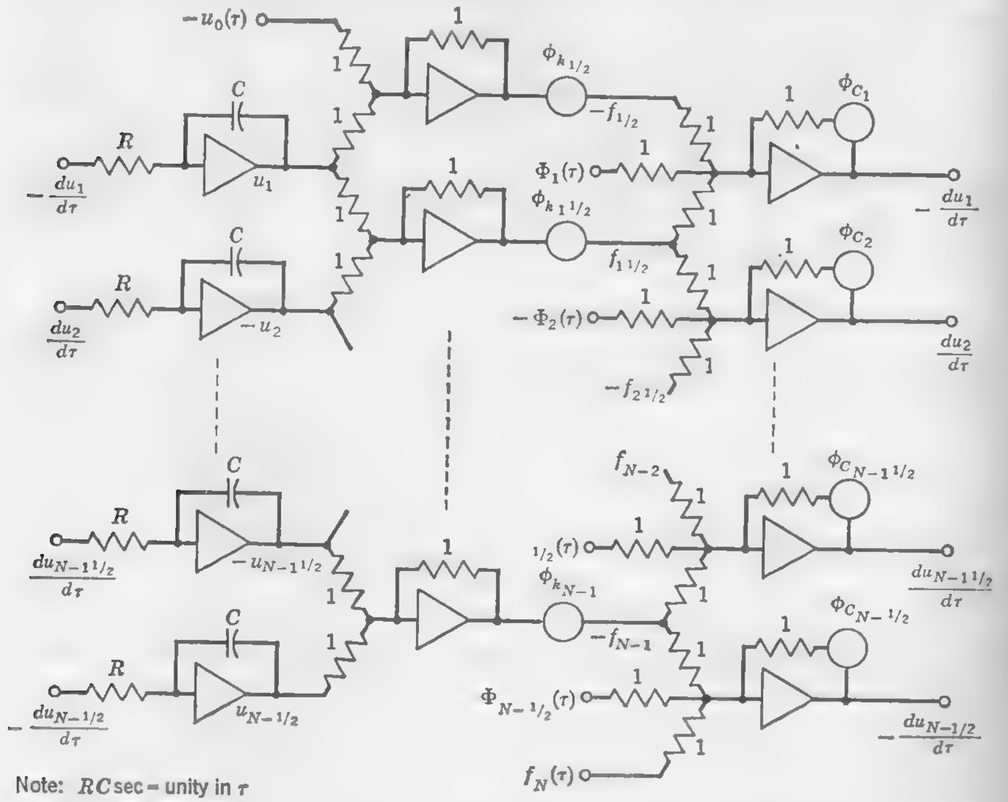


FIG. 5.10.10. Computer circuit for one-dimensional heat flow.

derivative, since x is fixed at $n\,\Delta x$ for u_n. Thus Eq. (41) is an ordinary differential equation. It is convenient to introduce a new dimensionless time variable τ given by

$$\tau = \frac{1}{(\Delta x)^2}t \tag{5.10.42}$$

where $\Delta x = 1/N$, there being N stations between $x = 0$ and $x = 1$. Then Eq. (41) becomes

$$\phi_{c_n}\frac{du_n}{d\tau} = \phi_{k_{n+\frac{1}{2}}}(u_{n+1} - u_n) - \phi_{k_{n-\frac{1}{2}}}(u_n - u_{n-1}) + \Phi_n(\tau) \tag{5.10.43}$$

where
$$\Phi_n(\tau) = (\Delta x)^2 S[n\,\Delta x, (\Delta x)^2 t] \tag{5.10.44}$$

Equation (43) is iterated for different values of n until the boundaries are reached. In the problem of Fig. 5.10.1 the left boundary temperature is zero. Thus $u_0 = 0$.

Actually, it adds no complexity to the computer circuit to let the boundary temperature be a prescribed time-dependent function $u_0(\tau)$. The right boundary is an insulator, so that $\left.\frac{\partial u}{\partial x}\right|_{x=1} = 0$. From Eq. (39) this implies that $u_r = u_{r-1}$, where $r - 1$ is the last station along $x(N = r - \frac{1}{2})$. Note that the insulator boundary occurs at a half-integral station. When Eq. (44) is iterated between the boundaries, we obtain $N - \frac{1}{2}$ simultaneous first-order differential equations. The electronic-analog-computer circuit is shown in Fig. 5.10.10, where we have defined a quantity $f_{n+\frac{1}{2}}$ proportional to the heat flux at the $n + \frac{1}{2}$ station and given by the formula

$$f_{n+\frac{1}{2}} = \phi_{k_{n+\frac{1}{2}}}(u_{n+1} - u_n) \tag{5.10.45}$$

At the right boundary $f_N = 0$ for our original problem. In general, as shown in Fig. 5.10.10, it can be prescribed time-varying function $f_N(\tau)$. Note that the output

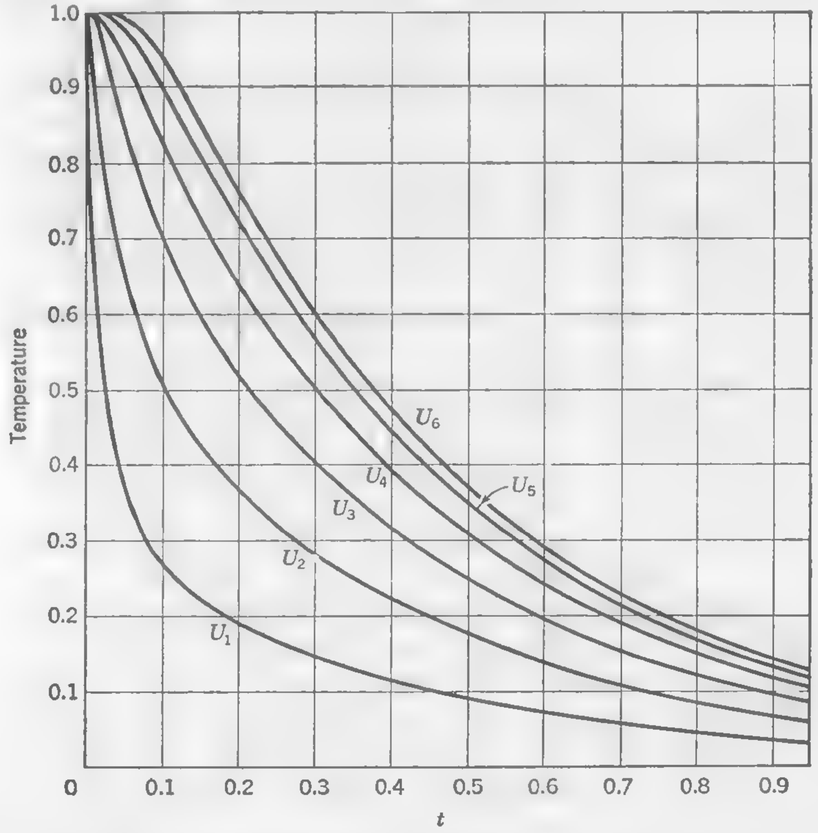


FIG. 5.10.11. Computer solution for transient heat flow in a slab.

of each successive row of amplifiers is reversed. This allows the necessary differences to be taken without adding inverting amplifiers. Note also that the quantity $f_{n+\frac{1}{2}}$, proportional to heat flux, is available at each half station. Single coefficient potentiometers represent the heat-capacity parameter ϕ_{c_n} and the conductivity parameter $\phi_{k_{n+\frac{1}{2}}}$ at each station. The external heat input Φ_n at each station is introduced as a time-varying input voltage. The integrator time scale for the circuit is RC sec, i.e., RC sec of real computer time represents one unit of dimensionless time τ. Initial conditions on the integrators represent initial temperatures at the different stations. Actual computer solutions for a slab with left edge at zero temperature, right edge insulated, and an initial temperature of unity across the slab are shown in Fig. 5.10.11 for $N = 6\frac{1}{2}$. Here the temperatures at each station are plotted as a function of time.

In the comparable result shown in Fig. 5.10.7, where the separation-of-variables method was used, the temperatures at a given time were plotted as a function of distance x through the slab.

If the slab is uniform, $\phi_{c_n} = 1$, $\phi_{k_{n+\frac{1}{2}}} = 1$, and Eq. (43) becomes

$$\frac{du_n}{d\tau} = u_{n+1} - 2u_n + u_{n-1} + \Phi_n(\tau) \tag{5.10.46}$$

On the right side we have the central-difference approximation to $[\partial^2 u/\partial x^2](\Delta x)^2$. Thus

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_n = \frac{u_{n+1} - 2u_n + u_{n-1}}{(\Delta x)^2} \tag{5.10.47}$$

The solution of Eq. (46) can be accomplished simply by using only one amplifier per station, as shown in Fig. 5.10.12 for the nth station. This can also be done for a nonuniform slab, but the circuit of Fig. 5.10.10 has the advantage of being a more direct analogy in this case.

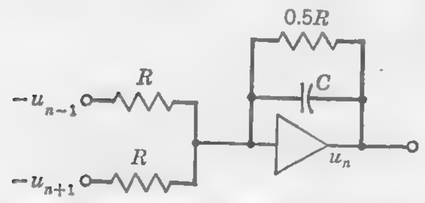


FIG. 5.10.12. Computer circuit for heat flow using one amplifier per station.

5.10.13. Estimated Accuracy of the Difference Method Using a Taylor-series Expansion. Since the replacement of partial derivatives by finite differences is only an approximation, it is important to estimate the errors inherent in the method. We shall use two ways of accomplishing this. First we shall examine the error terms by means of a Taylor series. Second, we shall separate variables for the difference-equation case and compare the resulting mode shapes and frequencies with the continuous values given in the previous section.

Consider the Taylor-series expansion for u_{n+1} in terms of $u_{n+\frac{1}{2}}$, where $\Delta x = h$. We have

$$u_{n+1} = u_{n+\frac{1}{2}} + \frac{h}{2}\left.\frac{\partial u}{\partial x}\right|_{n+\frac{1}{2}} + \frac{1}{2!}\left(\frac{h}{2}\right)^2\left.\frac{\partial^2 u}{\partial x^2}\right|_{n+\frac{1}{2}} + \frac{1}{3!}\left(\frac{h}{2}\right)^3\left.\frac{\partial^3 u}{\partial x^3}\right|_{n+\frac{1}{2}} + \cdots \tag{5.10.48}$$

Similarly

$$u_n = u_{n+\frac{1}{2}} - \frac{h}{2}\left.\frac{\partial u}{\partial x}\right|_{n+\frac{1}{2}} + \frac{1}{2!}\left(\frac{h}{2}\right)^2\left.\frac{\partial^2 u}{\partial x^2}\right|_{n+\frac{1}{2}} - \frac{1}{3!}\left(\frac{h}{2}\right)^3\left.\frac{\partial^3 u}{\partial x^3}\right|_{n+\frac{1}{2}} + \cdots \tag{5.10.49}$$

Subtracting Eq. (49) from Eq. (48) and solving for $\left.\frac{\partial u}{\partial x}\right|_{n+\frac{1}{2}}$, we have

$$\left.\frac{\partial u}{\partial x}\right|_{n+\frac{1}{2}} = \frac{u_{n+1} - u_n}{h} + \epsilon \tag{5.10.50}$$

where

$$\epsilon = -\frac{h^2}{24}\left.\frac{\partial^3 u}{\partial x^3}\right|_{n+\frac{1}{2}} + \cdots \tag{5.10.51}$$

Thus the error in the central-difference formula given in Eq. (39) varies as h^2 for small $h(h = \Delta x)$. If the number of stations is doubled, h is halved, and the error is reduced by a factor of 4. Using the same technique one can show that the central-difference approximation to $\left.\frac{\partial^2 u}{\partial x^2}\right|_n$ given by Eq. (47) has an error ϵ given by

$$\epsilon = -\frac{h^2}{12}\left.\frac{\partial^4 u}{\partial x^4}\right|_n + \cdots \tag{5.10.52}$$

Again the accuracy improves as h^2 for decreasing interval size h.

Later in this section we shall examine the error ϵ associated with forward- and backward-difference approximations in addition to the central-difference approximations given above. We shall also give ϵ for higher-order difference approximations. Now let us turn to the method of separation of variables for analysis of the errors.

5.10.14. Estimated Accuracy of the Difference Method Using Separation of Variables. The method of separation of variables can be used not only in the case of a continuous heat-flow problem but also in the case of our cellular approximation. In the latter case we shall obtain a number of discrete normal-made solutions, each characterized by a simple exponential decay in time. By comparing mode shapes and decay constants with their counterparts for the continuous-medium case, we can estimate the accuracy of the difference method. Consider the case shown in Fig. 5.10.1, where for simplicity we assume a uniform medium ($\phi_c = \phi_k = 1$). The solution to the eigenvalue problem given by Eq. (19) with boundary conditions of Eq. (21) is easily seen to be the following:

$$X_n(x) = \sin(n - \tfrac{1}{2})\pi x \qquad n = 1,2,3, \ldots \tag{5.10.53}$$

i.e.,

$$\beta_n = (n - \tfrac{1}{2})^2\pi^2 \tag{5.10.54}$$

Thus the complete solution for the nth mode becomes

$$u_n(x,t) = A_n[\sin(n - \tfrac{1}{2})\pi x]e^{-\beta_n t} \tag{5.10.55}$$

Let us assume that the sinusoidal spatial modes given by the above equation are also sinusoidal in the case of the difference-equation representation, i.e., assume that the temperatures at discrete intervals along x fall on a sinusoidal distribution. Then at the ith station $(x = i/N)$

$$u_{in}(t) = A_n\left[\sin\frac{(n - \tfrac{1}{2})\pi i}{N}\right]e^{-\bar{\beta}_n \tau/N^2} \tag{5.10.56}$$

Here $\bar{\beta}_n$ is the decay constant associated with the nth mode of the difference equation. Substituting Eq. (56) into Eq. (46) and letting $\Phi_n(\tau) = 0$, we find that the equation is satisfied provided that

$$\bar{\beta}_n = 2N^2\left[1 - \cos\frac{(n - \tfrac{1}{2})\pi}{N}\right] \tag{5.10.57}$$

After expansion of the cosine function in a power series we have

$$\bar{\beta}_n = (n - \tfrac{1}{2})^2\pi^2\left[1 - \frac{(n - \tfrac{1}{2})^2\pi^2}{12N^2} + \cdots\right] \tag{5.10.58}$$

In the limit of infinitely many cells the $\bar{\beta}_n$ given above converge to the β_n given in Eq. (54) for the continuous medium. The error in $\bar{\beta}_n$ varies as $1/N^2$ or h^2, as also seen in the error term in Eq. (52). In Fig. 5.10.13 are shown the percentage deviations in decay constant $\bar{\beta}_n$ for different mode numbers n as a function of the number of cells N.

To summarize, we see that, when the space derivatives of the one-dimensional heat equation in a uniform medium are replaced by finite differences, the resulting mode shapes agree exactly with those for the continuous solution, whereas the decay constants (eigenvalues) for each mode are somewhat smaller. This means that the higher modes will decay somewhat more slowly when the difference method is used. Note that instead of an infinite number of modes, there are only $N - \frac{1}{2}$ modes when the difference-equation approach is used. Note also from Fig. 5.10.13 that, if the first two or three modes are by far the most important, only 5 to 10 stations are needed for reasonable results.

5.10.15. Higher-order Central Differences and Backward Differences. By using Taylor expansions similar to Eqs. (48) and (49) one can develop higher-order and therefore more accurate central-difference approximations, as well as forward- and backward-difference approximations.[5] The latter approximate derivatives at a point

by including points on one side only. These various schemes, along with the first term of the error ϵ, are given in Table 5.10.1. Backward-difference formulas are symmetrical with the forward-difference formulas.

Note that the higher-order central-difference approximations, although more accurate, must be modified at the boundaries. Either a lower-order central difference is required there, or a higher-order backward or forward difference must be used. In

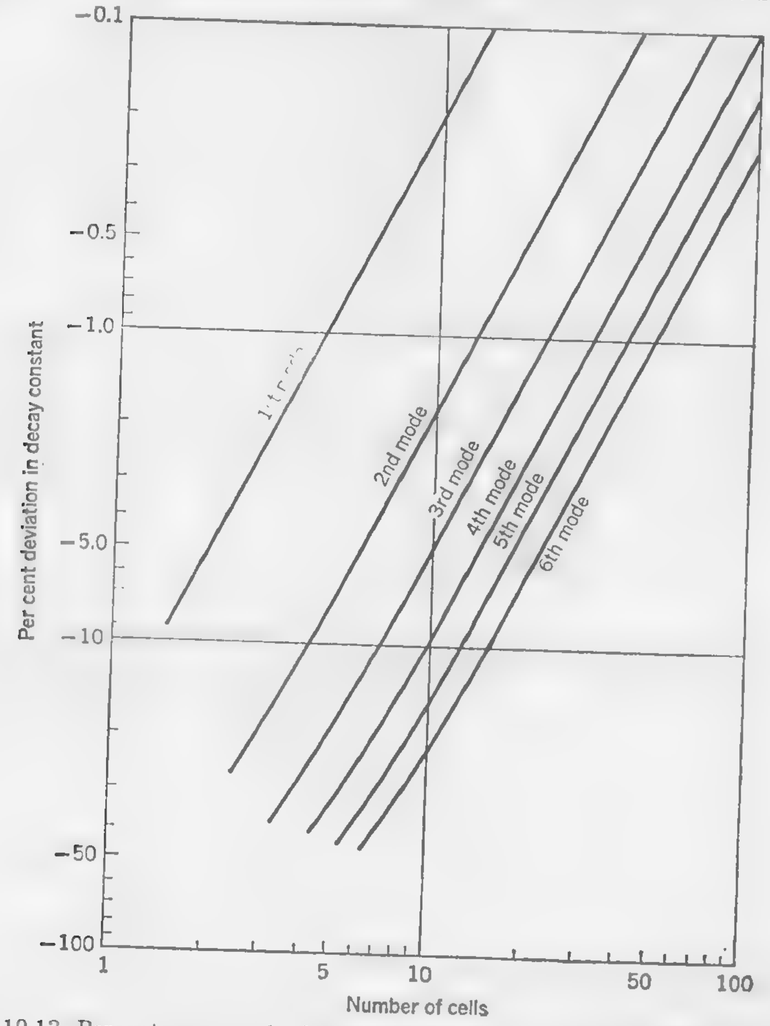


FIG. 5.10.13. Percentage error in decay constant $\bar{\beta}_n$ for one-dimensional heat flow.

any case, symmetry is destroyed. This problem can be avoided in certain cases by using the following second-order central-difference approximation, directly derivable from Taylor expansions:

$$\frac{1}{12}\frac{\partial^2 u}{\partial x^2}\bigg|_{n-1} + \frac{10}{12}\frac{\partial^2 u}{\partial x^2}\bigg|_{n} + \frac{1}{12}\frac{\partial^2 u}{\partial x^2}\bigg|_{n-1} = \frac{u_{n+1} - 2u_n + u_{n-1}}{h^2} \tag{5.10.59}$$

Here the error is of order h^4 but only the variable u_n and grid points immediately on either side, as well as $\left.\frac{\partial^2 u}{\partial x^2}\right|_n$ and grid points on either side, are required. At any station the terms can be computed implicitly through feedback loops from adjacent summers with output $\left.\frac{\partial^2 u}{\partial x^2}\right|_{n+1}$ and $\left.\frac{\partial^2 u}{\partial x^2}\right|_{n-1}$.

Table 5.10.1. Summary of Central and Forward Difference Approximations for First and Second Derivatives

Type of approximation	Approximation formula	First error term
Central difference, first order	$\left.\frac{\partial u}{\partial x}\right\|_n = \frac{u_{n+1} - u_{n-1}}{2h}$	$-\frac{h^2}{6}\left.\frac{\partial^3 u}{\partial x^3}\right\|_n$
Central difference, second order	$\left.\frac{\partial u}{\partial x}\right\|_n = \frac{-u_{n+2} + 8u_{n+1} - 8u_{n-1} + u_{n-2}}{12h}$	$\frac{h^4}{30}\left.\frac{\partial^5 u}{\partial x^5}\right\|_n$
Forward difference, first order	$\left.\frac{\partial u}{\partial x}\right\|_n = \frac{u_{n+1} - u_n}{h}$	$-\frac{h}{2}\left.\frac{\partial^2 u}{\partial x^2}\right\|_n$
Forward difference, second order	$\left.\frac{\partial u}{\partial x}\right\|_n = \frac{-u_{n+2} + 4u_{n+1} - 3u_n}{2h}$	$\frac{h^2}{3}\left.\frac{\partial^3 u}{\partial x^3}\right\|_n$
Central difference, first order	$\left.\frac{\partial^2 u}{\partial x^2}\right\|_n = \frac{u_{n+1} - 2u_n + u_{n-1}}{h^2}$	$-\frac{h^2}{12}\left.\frac{\partial^4 u}{\partial x^4}\right\|_n$
Central difference, second order	$\left.\frac{\partial^2 u}{\partial x^2}\right\|_n = \frac{-u_{n+2} + 16u_{n+1} - 30u_n + 16u_{n-1} - u_{n-2}}{12h^2}$	$\frac{h^4}{90}\left.\frac{\partial^6 u}{\partial x^6}\right\|_n$
Forward difference, first order	$\left.\frac{\partial^2 u}{\partial x^2}\right\|_n = \frac{u_{n+2} - 2u_{n+1} + u_n}{h^2}$	$-h\left.\frac{\partial^3 u}{\partial x^3}\right\|_n$
Forward difference, second order	$\left.\frac{\partial^2 u}{\partial x^2}\right\|_n = \frac{-u_{n+3} + 4u_{n+2} - 5u_{n+1} + 2u_n}{h^2}$	$\frac{11h^2}{12}\left.\frac{\partial^4 u}{\partial x^4}\right\|_n$

It has been pointed out that use of higher-order difference approximations allows spurious mode shapes not in the original problem.[6] Actually, these spurious modes will never appear because of the boundary constraints, and hence present no problem.

5.10.16. Heat Flow with Mixed Boundary Conditions. Often the boundary condition at one edge of a slab, say at $x = 0$, states that the flux is proportional to the temperature, i.e., $\frac{\partial u}{\partial x}(0) = ku(0)$. One way to treat this problem is to calculate $\left.\frac{\partial^2 u}{\partial x^2}\right|_0$ by means of a forward difference involving $\partial u/\partial x$ at stations 0, ½, and 1. Thus

$$\frac{du_0}{dt} = \left.\frac{\partial^2 u}{\partial x^2}\right|_0 = \frac{1}{2h}\left[-3\left.\frac{\partial u}{\partial x}\right|_0 + 4\left.\frac{\partial u}{\partial x}\right|_{1/2} - \left.\frac{\partial u}{\partial x}\right|_1\right] \tag{5.10.60}$$

where $$\left.\frac{\partial u}{\partial x}\right|_0 = ku_0 \qquad \left.\frac{\partial u}{\partial x}\right|_{1/2} = \frac{u_1 - u_0}{h} \qquad \left.\frac{\partial u}{\partial x}\right|_1 = \frac{u_2 - u_0}{2h} \tag{5.10.61}$$

5.10.17. Two-dimensional Heat Flow. Here the heat-flow equation in rectilinear coordinates $\bar{x}$ and $\bar{y}$ for a uniform medium becomes

$$C_0\frac{\partial u}{\partial \bar{t}} = K_0\left(\frac{\partial^2 u}{\partial \bar{x}^2} + \frac{\partial^2 u}{\partial \bar{y}^2}\right) \tag{5.10.62}$$

where, for simplicity, we have neglected any internal heat sources. As previously we introduce dimensionless distance variables $x = \bar{x}/L_x$, $y = \bar{y}/L_y$, where L_x and L_y are the rectangle dimensions in the $\bar{x}$ and $\bar{y}$ direction, respectively. Then by letting $t = (K_0/C_0L_x^2)\bar{t}$ and $\alpha_{xy} = L_x/L_y$, we have

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha_{xy}^2\frac{\partial^2 u}{\partial y^2} \tag{5.10.63}$$

Dividing the medium into N stations along x and y, as shown in Fig. 5.10.14, and letting $\tau = N^2t$, we have for the equilibrium equation at the ij station

$$\frac{du_{ij}}{d\tau} = u_{i+1,j} + u_{i-1,j} - 2(1 + \alpha_{xy}^2)u_{ij} + \alpha_{xy}^2u_{i,j+1} + \alpha_{xy}^2u_{i,j-1} \tag{5.10.64}$$

where we have used first-order central-difference approximations. Boundary conditions are imposed in a manner exactly analogous to the one-dimensional case. Error analysis by use of separation of variables shows that mode shapes for the cellular approximation in Fig. 5.10.14 agree exactly with the continuous case (i.e., are sinusoidal for homogeneous boundary conditions), while decay constants show percentage errors analogous to those in Fig. 5.10.13.

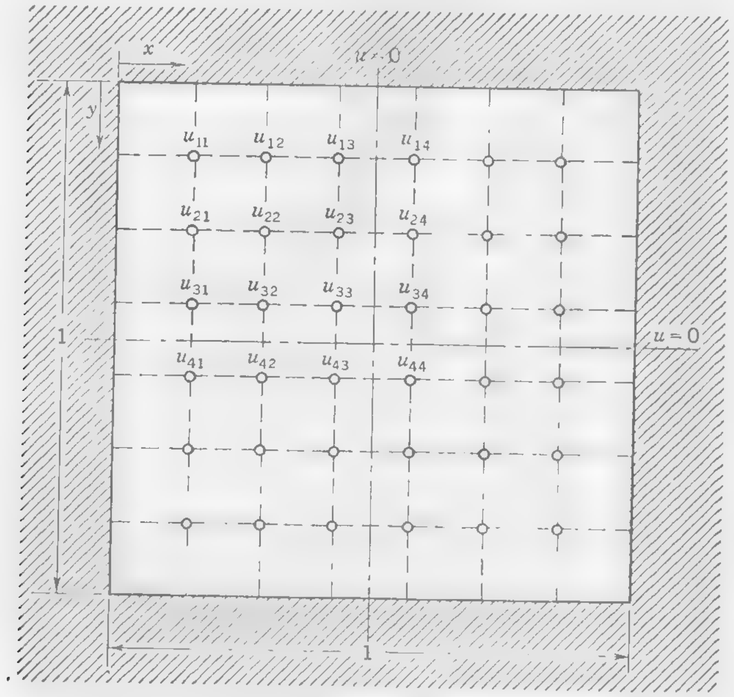


FIG. 5.10.14. Two-dimensional heat-flow problem.

5.10.18. Heat Flow in Cylinders. Frequently it is more convenient to use cylindrical rather than rectilinear coordinates because the boundary is cylindrical. For axially symmetric heat flow in an infinite rod the heat equation becomes[1]

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\, r \frac{\partial u}{\partial r} + S(r,t) \qquad (5.10.65)$$

where r is the dimensionless radial distance from the axis ($r = 1$ at the wall). Here $t = (K_0/C_0 r_0^2)\bar{t}$ and is dimensionless, where K_0 is the conductivity, C_0 the heat capacity, r_0 the cylinder radius, and $\bar{t}$ is real time. Because of axial symmetry we have the following boundary condition at $r = 0$:

$$\frac{\partial u}{\partial r}(0) = 0 \qquad (5.10.66)$$

At the wall ($r = 1$) the boundary condition can be specified on the temperature u, the flux $(1/r)(\partial u/\partial r)$, or any linear combination of these. To treat the problem using the difference method we consider u only at discrete intervals Δr. Thus at the nth station we have from Eq. (65)

$$\frac{du_n}{d\tau} = \frac{n + \frac{1}{2}}{n}(u_{n+1} - u_n) - \frac{n - \frac{1}{2}}{n}(u_n - u_{n-1}) + (\Delta r)^2 S_n(\tau) \qquad (5.10.67)$$

where $\tau = t/(\Delta r)^2$. To enforce the boundary condition of Eq. (66) we let $u_0 = u_1$, which implies that the axis ($r = 0$) corresponds to the ½ station. The boundary condition at the wall ($r = 1$) is enforced in the same manner that was used previously for the heat equation.

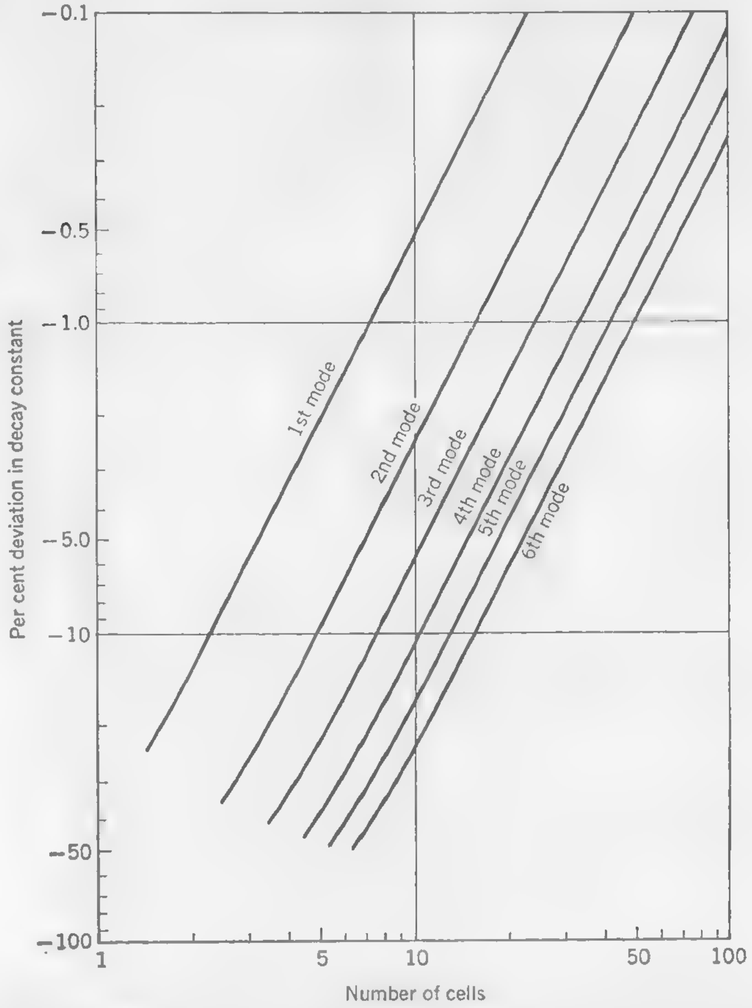


FIG. 5.10.15. Percentage error in decay constant $\bar{\beta}_n$ for radially dependent modes of cylindrical heat flow.

The error in the central-difference approximations in Eq. (67) varies as $(\Delta r)^2$ as previously. Comparison of normal-mode shapes with the exact solution for the continuous case shows approximate but not exact agreement.[7] Figure 5.10.15 shows the deviation of the decay constants (eigenvalues).

For heat flow which is not axially symmetric u depends on polar angle ϕ. For a finite-length rod it also depends on distance z along the rod. The resulting partial differential equation becomes

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\, r \frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \phi^2} + \alpha_{rz}^2 \frac{\partial^2 u}{\partial z^2} + S(r,\phi,z,t) \qquad (5.10.68)$$

where $\alpha_{rz} = r_0/z_0$, z_0 being the rod length. The difference method requires stations along ϕ and z as well as along r. The resulting equations are straightforward but

require considerable analog equipment. Using central-difference approximations the errors in ϕ- and x-dependent mode shapes are zero, but decay constants are somewhat lower than the exact values, much as shown in Fig. 5.10.13.

5.10.19. Spherical and Other Boundary Shapes. Equations similar to those in the previous section can be used for $\nabla^2 u$ in spherical coordinates.[7] For other boundary shapes, including irregular boundaries, methods of numerical analysis have been developed which can be used directly to obtain the difference-equation representation.[5]

5.10.20. Change of Variable to Group Stations Unevenly. Often the expected temperature variations are more severe in one region of the medium, e.g., the region near $x = 0$ in Figs. 5.10.7 and 5.10.11. By grouping the stations more closely in this region, higher accuracy should result. This can usually be done with a simple transformation. For example, in the problem of Fig. 5.10.7 let a new distance variable $y = \sqrt{x}$. If we rewrite the heat equation in terms of y and take equally spaced stations along y, this will in effect give us much closer spacing near $x = 0$ in the x stations.

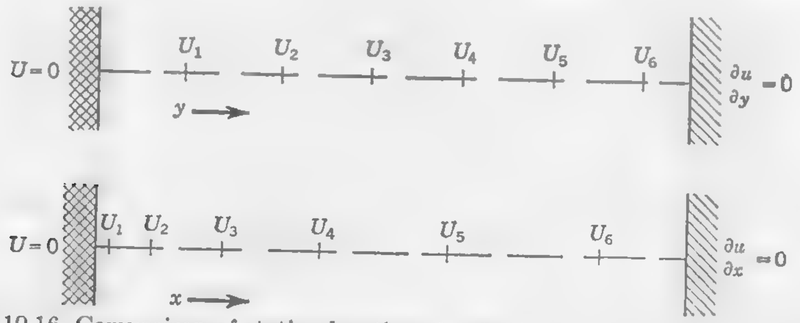


FIG. 5.10.16. Comparison of station locations for the space variable transformation.

This is illustrated in Fig. 5.10 16, which compares the station locations in the two cases when $N = 6.5$. We define the new distance variable y as

$$y = x^{1/2} \qquad \frac{\partial}{\partial x} = \frac{\partial}{\partial y}\frac{dy}{dx} = \frac{1}{2y}\frac{\partial}{\partial y} \tag{5.10.69}$$

For a uniform medium and $S = 0$ Eq. (10) becomes, in terms of y,

$$\frac{\partial u}{\partial t} = \frac{1}{2y}\frac{\partial}{\partial y}\left[\frac{1}{2y}\frac{\partial u}{\partial y}\right] \tag{5.10.70}$$

After division of y into N stations the difference equation can be written using central-difference approximations. The resulting analog-computer solution yields accurate temperature solutions much closer to $x = 0$ than previously obtained in Fig. 5.10.11.[7] There are many suitable transformations which can be used to change station grouping.

5.10.21. Solution of Heat-flow Problems in Semi-infinite Media. Consider a heat-flow problem between $x = 0$ and $x = \infty$. Here an infinite number of stations would be required if equal increments between stations were used. However, by employing the transformation $y = 1 - e^{-x}$ the region $0 < x < \infty$ is mapped into the region $0 < y < 1$. Then equally spaced stations in y can be used to set up the difference equations.[7,8] For this technique to be useful the solution u should approach a constant value as $x \to \infty$. Thus it cannot be used in the case of the wave equation, where the solution would oscillate as $x \to \infty$.

5.10.22. Nonlinear Heat-flow Problems. When the difference method is used, there is no reason why the analog computer cannot be used to solve nonlinear partial differential equations. For example, the conductivity K may be a function of temperature u. This nonlinearity is simply introduced at each station.[7,8] This does mean, however, that each nonlinearity in the original equation requires N nonlinearities in the computer circuit, where N is the number of active stations.

5.10.23. Problems with Moving Boundary Conditions. An example of this type of problem is the melting or oblating problem. It can be solved with the analog computer by introducing a new distance variable $y = x/l(t)$, where $l(t)$ is the position of the moving boundary in x. Whereas $0 < x < l(t)$, $0 < y < 1$ and the boundary remains fixed in y. Care must be taken in writing the transformed equations (nonlinearities are introduced because of the transformation). Stations are then taken along y and the difference equations set up for the analog computer.[9]

5.10.24. Solution of the Wave Equation. Reference to Eq. (2) shows that in general the wave equation is similar to the heat equation except that it involves the second time derivative rather than the first. Thus all the discussion in the preceding articles regarding difference techniques for approximating spatial derivatives applies to the wave equation as well as the heat equation. The computer circuit at each station will require two integrators instead of one, as in Fig. 5.10.10, but a total of three amplifiers per station is still adequate.[8,10]

5.10.25. Transverse Displacement of Thin Beams. The equation for the transverse displacement $y(x,t)$ of a thin beam arises frequently in structural-dynamics problems. If we consider the displacement only at discrete, equally spaced stations along the beam, then from Eq. (27) we have for the equilibrium of forces at the nth station

$$\phi_{d_n}\frac{d^2 y_n}{d\tau^2} = -m_{n+1} + 2m_n - m_{n-1} + \Phi_n(\tau) \tag{5.10.71}$$

where m_n is proportional to the bending-moment at the nth station and is given by

$$m_n = \phi_{f_n}(y_{n+1} - 2y_n + y_{n-1}) \tag{5.10.72}$$

Here τ is a new dimensionless time equal to $t/(\Delta x)^2$ and $\Phi_n(\tau)$ is given by

$$\Phi_n(\tau) = (\Delta x)^4 f[nx, (\Delta x)^2\tau] \tag{5.10.73}$$

First-order central-difference approximations have been used here. Equations (71) and (72) are iterated at each station until the boundaries are reached. There are a number of possible ways to approximate the boundary conditions, but Table 5.10.2 presents the most widely used technique and one which is known to yield accurate results.[2,10]

Table 5.10.2. Boundary Conditions Using Difference Approximations for Various End Fastenings of Beams

Type of end fastening	Where end occurs	Boundary condition
Hinged (simply supported).....	N	$y_N = m_N = 0$
Built-in.....	$N + \frac{1}{2}$	$y_N = y_{N+1} = 0$
Free.....	$N + \frac{1}{2}$	$m_N = m_{N+1} = 0$

The electronic-analog-computer circuit for a cantilever beam with the end at $x = 0$ (station $\frac{1}{2}$) built in and the end at $x = 1$ (station $N + \frac{1}{2}$) free is shown in Fig. 5.10.17. Note that three amplifiers per station are required. In order to make the time scale reasonable, particularly for large $N(\tau = N^2 t)$ it is usually desirable to let the integrator time scale $RC = 0.1$ sec or less.

Since we have used first-order central differences in writing Eqs. (71) and (72), the error should vary as $(\Delta x)^2$, i.e., as $1/N^2$. A more meaningful check is obtained by comparing the normal-mode shapes and frequencies of the cellular beam with those for the continuous beam. This has been done for uniform beams[2,10] with the following results: mode shapes for hinged-hinged beams agree exactly, for free-free beams almost exactly, and for clamped-clamped and clamped-free (cantilever) beams fairly accurately, depending of course on the number of stations N. Percentage deviation in normal-mode frequency for cantilever, free-free, and clamped-clamped beams is shown in Fig. 5.10.18. Note that with only six stations (15 operational amplifiers,

since $y_{\frac{1}{2}} = 0$) the first three mode frequencies of a cantilever beam are accurate to 1.5, 1.0, and −5.6 per cent, respectively. Since any beam motion can be represented as a superposition of its normal modes, this implies that a six-station representation can be reasonably accurate.

5.10.26. Displacement of Thick Beams Including Transverse Shear Effects. In this case the beam slope $\partial y/\partial x$ includes a contribution due to transverse shear.[3] As a result the equation for the bending moment M includes a term proportional

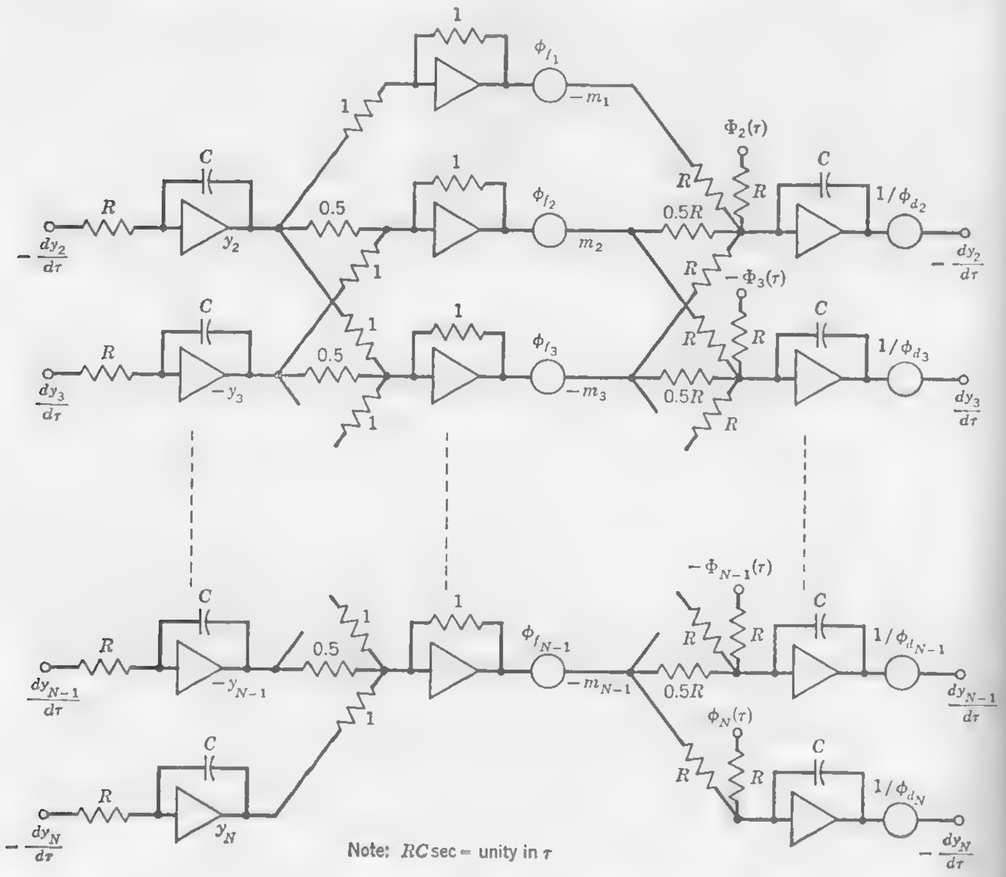


FIG. 5.10.17. Computer circuit for N-cell cantilever beam.[1]

to $\partial^2 M/\partial x^2$ as well as the usual $\partial^2 y/\partial x^2$. Then Eq. (72) for m_n is replaced by the equation[2]

$$m_n = \phi_{f_n}(y_{n+1} - 2y_n + y_{n-1}) + \left(\frac{EI}{kAG}\right)_n \frac{1}{(L\,\Delta x)^2}(m_{n+1} - 2m_n + m_{n-1}) \quad (5.10.74)$$

where EI is the stiffness, kAG the shear rigidity, and L the beam length. In general six amplifiers per station are needed for this representation and a feedback loop containing only summing amplifiers is present at each station. The inclusion of the transverse shear effect can become important for short thick beams. Also, rotary-inertia effects can become important under these conditions.[3,11]

5.10.27. Beams with Damping. Both linear (viscous) and nonlinear damping can be included in the analog-computer circuit for the beam.[2,12] Addition of viscous damping in Fig. 5.10.17 is accomplished simply by adding feedback around the integrators.

5.10.28. Stability of Beam Circuits. In general, the more stations one includes in a beam circuit, the larger the effect of any amplifier unbalance or noise. Also, free-free beams can translate and rotate in addition to vibrate, and the computer voltages repre-

senting displacements along the beam will tend to drift accordingly. Taking the fourth derivative, $\partial^4 y/\partial x^4$, all in one central-difference operation instead of two successive second-derivative operations is apt to lead to large errors due to summing-amplifier component inaccuracies. This is because of the small differences involved. When many stations N are used, the frequency of the lowest mode becomes very low indeed (e.g., 0.0156 cps for a six-cell cantilever beam if integrator time constants of

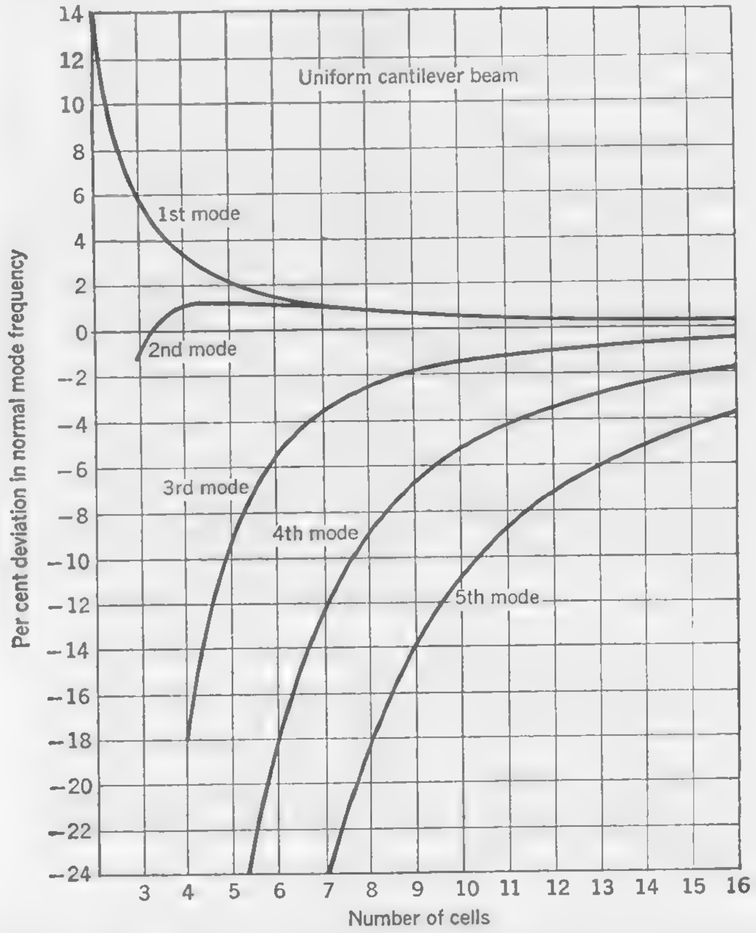


FIG. 5.10.18. Percentage error in normal-mode frequency for uniform cellular beams. First-, second-, third-, and fourth-mode error curves for free-free and clamped-clamped beams correspond to second-, third,- fourth-, and fifth-mode error curves, respectively, for the cantilever beam as shown.

1 sec are used). Thus one would like to use short integrator time constants. However, the highest mode may then be fairly high in frequency (23.2 cps for a six-cell beam with $RC = 0.025$ sec) and unless damping is added purposely, the highest mode may diverge because of limited computer bandwidth. The problem is aggravated by a factor N^2, where N is the number of stations along the beam. As a result of all these considerations the following conclusion is evident: never use more stations than appear necessary to realize the accuracy desired.

5.10.29. Representation of Plates and Built-up Structures. The equation representing the transverse displacement of thin plates is similar to Eq. (27) for thin beams except that the ∇^4 operator involves two dimensions instead of one.[13] Thus a two-dimensional grid is needed to yield the station locations for the finite-difference approximations to the spatial derivatives. A number of references summarize the actual difference equations and the method of approximating boundary conditions for clamped, free, and simply supported edges.[5,8,14,15]

More complicated complex structures, such as built-up multicellular structures, can be represented not only by passive-circuit analogs, as in Sec. 9.1, but also by electronic-analog-computer circuits.[16]

5.10.30. Comparison of Passive-circuit Methods and Electronic-analog-computer Techniques. Passive-circuit analogs for the *heat equation* are apt to be relatively inexpensive because they involve only resistors and capacitors. For this reason one would probably prefer them to active circuits if a special-purpose computer for solving linear heat-flow problems were needed, particularly for very large two- and three-dimensional problems. On the other hand, an active-circuit analog computer may already be available as part of a computing facility and would allow convenient incorporation of nonlinearities.

In the case of problems such as the wave equation, beam equation, and built-up structures a passive-circuit machine with high-quality components is apt to be appreciably more expensive than an electronic analog computer with the same problem capacity. In this case the chief advantage of the passive-circuit analog lies in the direct correspondence of its circuit elements to structural elements and the speed with which solutions can be generated. The operator of such a machine must be skilled in both the passive-circuit analogies and the physical problem being solved in order to utilize the full accuracy capability of the computer. On the other hand, the general-purpose electronic analog computer, if somewhat less tailor-made for particular problems, does have the advantages of high accuracy and versatility.

References

1. Churchill, R. V.: *Fourier Series and Boundary Value Problems*, McGraw-Hill Book Company, Inc., New York, 1941.
2. Howe, C. E., and R. M. Howe: Application of Difference Techniques to the Lateral Vibration of Beams Using the Electronic Differential Analyzer, University of Michigan Department of Aeronautical Engineering, Rept. AIR-7, 1954.
3. Timoshenko, S.: *Vibration Problems in Engineering*, D. Van Nostrand Company, Inc., Princeton, N.J., 1955.
4. Bisplinghoff, R. L., H. Ashley, and R. L. Halfman: *Aeroelasticity*, Addison-Wesley Publishing Company, Reading, Mass., 1955.
5. Salvadori, M. G., and M. L. Baron: *Numerical Methods in Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1952.
6. Fisher, M. E.: Higher Order Differences in the Analogue Solution of Partial Differential Equations, *J. Assoc. Computing Machinery*, **3**:325–347 (1956).
7. Howe, R. M.: Application of Difference Techniques to Heat Flow Problems Using the Electronic Differential Analyzer, University of Michigan Department of Aeronautical Engineering, Rept. AIR-10, 1954.
8. Karplus, W. J., and W. J. Soroka: *Analog Methods*, McGraw-Hill Book Company, Inc., New York, 1959.
9. Manci, O. J.: Differential Analyzer Solution of Diffusion Equation with a Free Boundary, Ph.D. Thesis, Instrumentation Engineering, University of Michigan.
10. Howe, R. M., and V. S. Haneman: Solution of Partial Differential Equations by Difference Techniques Using the Electronic Differential Analyzer, *Proc. IRE*, **41**: 1497–1508 (October, 1952).
11. Howe, C. E., and R. M. Howe: Application of the Electronic Differential Analyzer to the Oscillation of Beams Including Shear and Rotary Inertia, *J. Appl. Mechanics*, **22**:13–19 (1955).
12. Howe, R. M.: Electronic Differential Analyzer Solution of Beams with Nonlinear Damping, University of Michigan Department of Aeronautical Engineering, Rept. AIR-8, 1954.
13. Timoshenko, S.: *Theory of Plates and Shells*, McGraw-Hill Book Company, Inc., New York, 1940.
14. Williams, D., and P. C. Birchall: Solution of the Problem of the Square Cantilever Plate, Royal Aircraft Establishment, Farnborough, *Tech. Note*, Structures, 178, M.S. 25, March, 1956.
15. MacNeal, R. H.: The Solution of Elastic Plate Problems by Electrical Analogies, *J. Appl. Mechanics*, **18**:59–67 (1951).
16. Clymer, A. B.: *Handbook of Equations and Operational Analog Circuits for the Solution of Structural Problems*, North American Aviation, Inc., Columbus Division, 1950.

PART 11: SOLUTION OF ALGEBRAIC EQUATIONS, LINEAR PROGRAMMING, AND PARAMETER OPTIMIZATION

By L. D. Kovach *and* H. F. Meissinger

ALGEBRAIC EQUATIONS

5.11.1. Linear Algebraic Equations and Eigenvalue Problems. The solution of systems of linear algebraic equations of high order is important in many phases of scientific and engineering work and has stimulated the development of numerous methods employing analog computation. Many of the early mechanical and electrical simultaneous-equation solvers involving hand-driven gears, balanced tilting plates, and systems of transformers or potentiometers have been described by Soroka.[1] Most of these systems are operated by a sequence of manual adjustments which are continued until balance of all terms occurring in the system of equations is accomplished.

The electronic analog computer, owing to its flexibility and the ease of performing summation and multiplication of variables by constant coefficients, is well suited to this application. By the use of the feedback amplifier as a computing element, loading effects and low voltage levels can be avoided and the process of balancing the system of equations may be carried out automatically. However, with many feedback loops interacting in the computing setup, there is the possibility of unstable oscillations produced by positive feedback or by the accumulation of small phase shifts. Methods of resolving the stability problem, and hence of making the electronic analog computer generally applicable to the solution of simultaneous equations, have been described.[1 to 9]

To solve a system of n equations

$$\sum_{j=1}^{n} a_{ij}x_j - b_i = 0 \qquad i = 1, \ldots, n \tag{5.11.1}$$

one may start with arbitrary values for the x_j's. Thus one has

$$\sum_{j=1}^{n} a_{ij}x_j - b_i = \epsilon_i \qquad i = 1, \ldots, n \tag{5.11.2}$$

where ϵ_i is the *error* in the ith equation. The problem is to adjust the x_j's so as to satisfy Eq. (5.11.1) by using the ϵ_i's of Eq. (5.11.2).

A computer setup consisting of summing amplifiers and scale-factor potentiometers in which the first equation is solved for x_1, the second equation for x_2, etc., may tend to break into unstable high-frequency oscillations. To avoid this difficulty the system of algebraic equation (5.11.1) is rewritten as a set of differential equations

$$\begin{aligned} \frac{dx_1}{dt} + a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ \frac{dx_2}{dt} + a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \text{ etc.} \end{aligned} \tag{5.11.3}$$

which yield the constant values $x_1, x_2, \ldots, x_n$, as steady-state solutions, provided the real parts of the roots of the characteristic equation

$$\begin{vmatrix} a_{11}+s & a_{12} \cdots & a_{1n} \\ a_{21} & a_{22}+s \cdots & a_{2n} \\ a_{n1} & a_{n2} \cdots & a_{nn}+s \end{vmatrix} = 0 \tag{5.11.4}$$

are negative. Otherwise the system (5.11.3) yields unstable transient solutions $x_1(t)$, $x_2(t)$, . . . ,$x_n(t)$. To obtain stable solutions, one may distribute the derivatives in Eqs. (5.11.3) in different order or multiply some of the equations by -1. Another way to assure stability is to use a transformed set of equations obtained from the original set by multiplication with its transposed matrix. Then the equations to be instrumented on the computer are

$$\begin{aligned} \frac{dx_1}{dt} + a_{11}'x_1 + a_{12}'x_2 + \cdots + a_{1n}'x_n &= b_1' \\ \cdots\cdots\cdots\

then the relationships to be instrumented on the analog computer are

$$\left.\begin{aligned} &\sum_{j=1}^{n} a_{ij}x_j - b_i = \epsilon_i \\ &x_j = -\sum_{i=1}^{n} a_{ij}E_i \\ &\frac{dE_i}{dt} = K\epsilon_i \end{aligned}\right\} \quad i, j = 1, 2, \ldots, n \tag{5.11.8}$$

The method has the advantage of always producing stable solutions which converge to the appropriate steady-state values $x_1, x_2, \ldots, x_n$, provided that the algebraic system has a unique solution (i.e., the matrix of the coefficients is nonsingular). No transformation of the original set of equations need be performed in this case, which makes this method preferable to all others in the most general case.

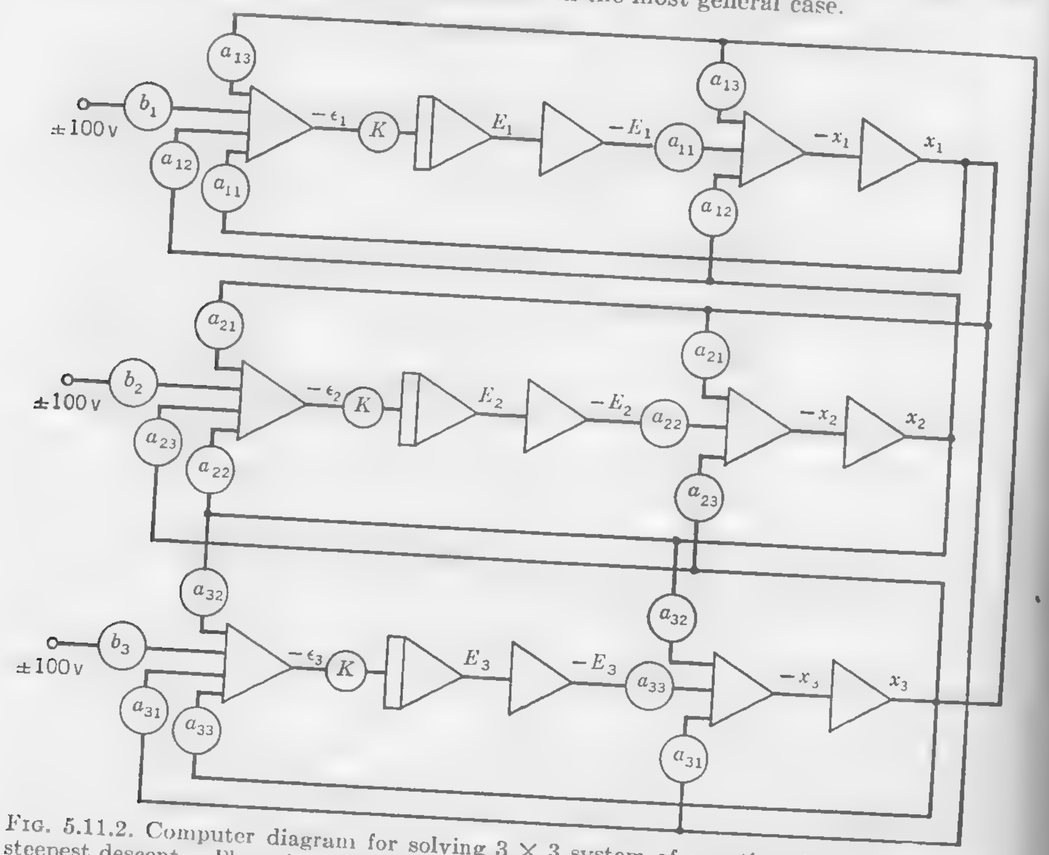


FIG. 5.11.2. Computer diagram for solving 3 × 3 system of equations by the method of steepest descent. Phase inverters are used to obtain negative coefficients a_{ik}.

To illustrate the method, a computer setup for solving the system (5.11.8) for the case $n = 3$ is shown in Fig. 5.11.2. Since each coefficient a_{ij} appears twice in this instrumentation, it is desirable for convenient operation to use ganged potentiometers for the pairs of coefficients.

It has been shown by Wilbur[11] that adjusting the variables x_j in a cyclic manner so as to reduce continuously the quantity μ of Eq. (5.11.5) will always cause μ to converge to zero. This, of course, occurs only when all the ϵ_i's are zero and hence the x_j have reached their correct values.

In the event that the solutions from the computer for a linear system are not as precise as desired, the following iterative process based on the superposition principle can be used to extend the accuracy. The values of x_i obtained from the computer may be substituted into the original set of equations and the residues r_i computed for each element (on a desk calculator, for example). Inserting these residues into the computer in place of the originally used constants b_i produces variational equations of the form

$$a_{i1}\,\Delta x_1 + a_{i2}\,\Delta x_2 + \cdots + a_{in}\,\Delta x_n = r_i \qquad i = 1,2, \ldots, n \tag{5.11.9}$$

The Δx_i are correction terms which, when combined with the original solution $x_i^{(0)}$, yield the more accurate solution

$$x_i^{(1)} = x_i^{(0)} + \Delta x_i \tag{5.11.10}$$

It is advantageous to multiply the r_i by a constant to utilize as large a scale factor as possible. If still higher accuracy is desired, the iteration process may be repeated.

Frequently a system of algebraic equations is given in the form

$$\sum_{j=1}^{n} (a_{ij} - b_{ij}\lambda)x_j = 0 \qquad i = 1,2, \ldots, n \tag{5.11.11}$$

If this system is to have solutions other than the trivial solution

$$x_1 = x_2 = \cdots = x_n = 0$$

it is necessary and sufficient that the determinant of the coefficient matrix vanish, that is,

$$\begin{vmatrix} (a_{11} - b_{11}\lambda) & (a_{12} - b_{12}\lambda) & \cdots & (a_{1n} - b_{1n}\lambda) \\ \vdots & \vdots & & \vdots \\ (a_{n1} - b_{n1}\lambda) & (a_{n2} - b_{n2}\lambda) & \cdots & (a_{nn} - b_{nn}\lambda) \end{vmatrix} = 0 \tag{5.11.12}$$

For a given set of coefficients a_{ij}, b_i, this is the case if, and only if, the parameter λ satisfies an equation of nth degree—the secular equation of the system—which is obtained by expanding the determinant (5.11.12). Hence, exactly n values of λ can be found which are known as the eigenvalues of the system. Adcock[8] and Gephart[9] have described methods of finding eigenvalues, without expanding the determinant, by a direct application of the simultaneous-equation solver. These apply primarily to systems having symmetrical matrices, i.e., those for which the eigenvalues λ are real. Gephart's method is indicated by an example. A system of third order is considered here, for which an augmented set of equations

$$\begin{aligned} t + \qquad\quad x_1 \qquad\quad + \qquad\quad & \qquad\quad = \text{const} \neq 0 \\ t + (a_{11} - \lambda)x_1 + \quad a_{12}x_2 + \quad & a_{13}x_3 = 0 \\ a_{21}x_1 + (a_{22} - \lambda)x_2 + \quad & a_{23}x_3 = 0 \\ a_{31}x_1 + \quad a_{32}x_2 + (a_{33} - \lambda)x_3 & = 0 \end{aligned} \tag{5.11.13}$$

is set up on the computer and solved for the unknown quantity t. During this process the parameter λ is slowly varied until a condition is found where $t = 0$. The value of λ giving rise to this solution is an eigenvalue. The search for other eigenvalues must then be continued.

If the linear algebraic system has complex coefficients, the methods of this section may be used except that the order of the system is doubled because it must be separated into real and imaginary parts.

5.11.2. Evaluation of Determinants. A method for evaluation of determinants, which makes use of the comparative ease of solving simultaneous linear equations on the analog computer, was devised by Roberson and McCool.[12]

Let A be a determinant of nth order defined by

$$A = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix} \tag{5.11.14}$$

This determinant may be interpreted as characterizing the linear system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= 1 \end{aligned} \tag{5.11.15}$$

where the terms on the right-hand side are zero with the exception of the last one, assumed to be 1. Therefore

$$x_n = \frac{1}{A}\begin{vmatrix} a_{11} & \cdots & 0 \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & 1 \end{vmatrix} = \frac{A_{n-1}}{A} \tag{5.11.16}$$

so that now A is expressed in terms of the solution x_n and a determinant of order $n - 1$, the latter being the cofactor of the term a_{nn} in the original determinant A. This process may then be continued by finding corresponding expressions

$$X_{n-1} = \frac{A_{n-2}}{A_{n-1}}, \qquad X_{n-2} = \frac{A_{n-3}}{A_{n-2}}, \qquad \cdots \tag{5.11.17}$$

thereby introducing determinants A_{n-2}, A_{n-3}, . . . of decreasing order, until the last remaining determinant A_1 is the known constant a_{11}. In this sequence the determinant, or cofactor, of order k is the one formed by the elements of the first k rows and columns of the original determinant. Finally, by combining the X_i on the equation solver, one obtains the relationship

$$x_n X_{n-1} X_{n-2} \cdots X_2 = \frac{A_{n-1}}{A}\frac{A_{n-2}}{A_{n-1}}\frac{A_{n-3}}{A_{n-2}} \cdots \frac{A_1}{A_2} = \frac{a_{11}}{A} \tag{5.11.18}$$

which yields A in terms of these X_i, x_n, and a_{11}.

A practical procedure for the computing sequence is to solve, first, the system of $n \times n$ equations, then to disconnect one equation after another each time a value X_i has been observed, and at the same time to change the input values, making the last entry on the right-hand side equal to unity in each case. The accuracy of this method of evaluation of determinants is estimated by Roberson and McCool as better than 10 per cent, depending, of course, on the structure of the determinant and the quality of the computer components. The question of stability in solving the sequence of equation systems of order n, $n - 1$, $n - 2$, . . . is of no concern as long as the general method (described previously) of stabilizing the equation solver is applied.

5.11.3. Inversion of Matrices. The problem of matrix inversion is of importance in many applications involving linear systems, but, except for systems of low order, the numerical procedure is a laborious one. An automatic equation solver which determines the roots x_j of the linear system

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad i = 1,2, \ldots ,n \tag{5.11.1}$$

can also be used to determine the elements c_{ij} of the inverse matrix.

If we use matrix notation, where (5.11.1) assumes the form $Ax = b$, and let

$$A^{-1} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \tag{5.11.19}$$

then Eqs. (5.11.1), when solved for x_j, yield $x = A^{-1}b$, or

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix} = \begin{bmatrix} c_{11}b_1 + c_{12}b_2 + \cdots + c_{1n}b_n \\ c_{21}b_1 + c_{22}b_2 + \cdots + c_{2n}b_n \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ c_{n1}b_1 + c_{n2}b_2 + \cdots + c_{nn}b_n \end{bmatrix} \tag{5.11.20}$$

Hence, if we solve the original system (5.11.1) successively for

$$b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \tag{5.11.21}$$

which are the columns of the identity matrix, we generate the columns of the inverse matrix.

It can be seen that, in order to invert an $n \times n$ matrix, it is necessary to solve an $n \times n$ system n times. The required changes in the computer setup are, however, relatively simple.

ANALOG-COMPUTER METHODS FOR FINDING ROOTS OF POLYNOMIAL EQUATIONS

5.11.4. General. The analog-computer techniques for determining the roots of polynomials fall into three principal categories from the point of view of computer operation: scanning, nulling, and tracking. In the *scanning technique* the computer sweeps the complex plane in a predetermined manner and locates the points where the polynomial vanishes. In the *nulling technique* the computer solves the polynomial by finding a path leading to a root such that a suitably defined error function is reduced to zero. Which root is determined depends on the point in the complex plane at which the process is started. Finally, in the *tracking mode*, the computer indicates the variation of a root that was previously determined by a nulling procedure, if one or several coefficients of the polynomial are being changed. The various techniques for finding roots of polynomials are illustrated and summarized in Fig. 5.11.3. Note that there is some overlap between these methods.

Before discussing methods for scanning the complex plane in search of roots z_i, some relationships used for convenient expression of a polynomial on analog computing equipment will be considered. Let a polynomial equation of the nth degree in z be given by

$$w(z) = a_n z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 = 0 \tag{5.11.22}$$

where $z = x + iy = re^{i\theta}$

The coefficients a_k shall be assumed to be real-valued. Separation of Eq. (5.11.22) into real and imaginary parts can be effected by expanding the binomials appearing in

$$w = u + iv = a_n(x + iy)^n + a_{n-1}(x + iy)^{n-1} + \cdots + a_1(x + iy) + a_0 = 0 \tag{5.11.23}$$

and collecting appropriate terms. This yields two equations of nth degree in the real variables x and y, which must be solved simultaneously by the computer. As an alternative, the use of polar coordinates r and θ and separation of (5.11.22) into real and imaginary parts yields

$$u = \sum_{k=0}^{n} a_k r^k \cos k\theta \tag{5.11.24}$$

$$v = \sum_{k=0}^{n} a_k r^k \sin k\theta \tag{5.11.25}$$

Expressions (5.11.24) and (5.11.25) follow immediately from Eq. (5.11.22) for polynomials of any degree and lend themselves to simple computer representation, whereas

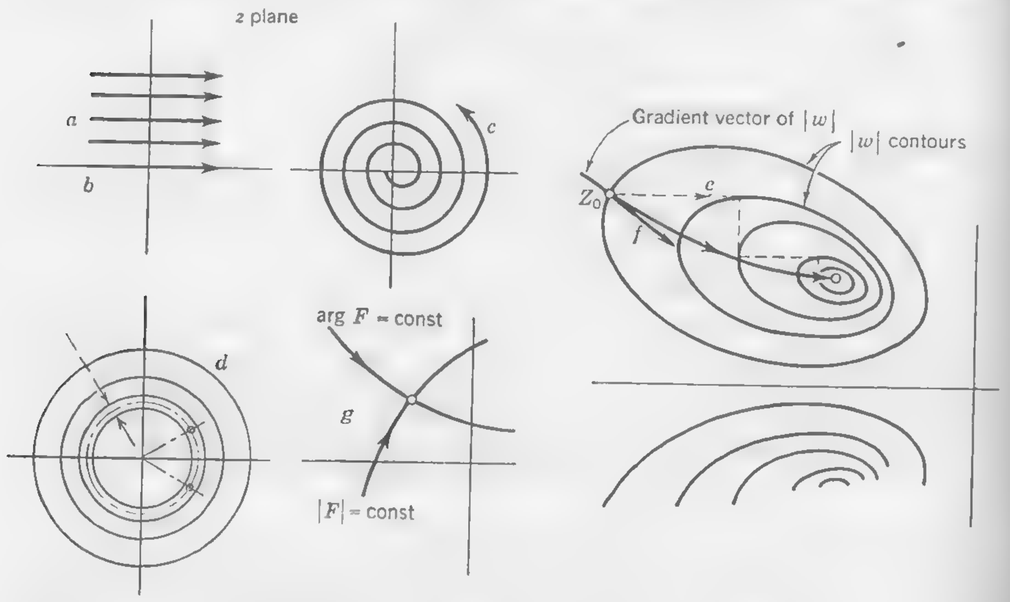


FIG. 5.11.3. Computer techniques for determining roots of polynomials.
Scanning: Linear scan (*a*), special case: real-axis scan (*b*), spiral scan (*c*), circular scan—Isograph (*d*), loci of constant modulus and argument (*g*).
Nulling: Bracketing—Isograph (*d*), relaxation (*e*), loci of constant modulus and argument (*g*), steepest descent (*f*).
Tracking: Loci of constant modulus and argument (*g*), steepest descent (*f*).

the separation of real and imaginary parts in Eq. (5.11.23) requires considerable manipulation if n is large, and gives rise to involved computer circuits. Hence, the procedure employing Eqs. (5.11.24) and (5.11.25), also known as harmonic synthesis, enjoys a more widespread use.

To illustrate the different requirements of the two procedures in terms of computing equipment, the case of a polynomial of fourth degree is considered here. The use of cartesian coordinates yields the equations

$$u = a_4x^4 + a_3x^3 + a_2x^2 + a_1x - 3a_3xy^2 - 6a_4x^2y^2 - a_2y^2 + a_4y^4 + a_0 = 0 \tag{5.11.26}$$

$$v = 4a_4x^3y + 3a_3x^2y + 2a_2xy - 4a_4xy^3 - a_3y^3 + a_1y = 0 \tag{5.11.27}$$

A large amount of multiplying equipment is required if both x and y are treated as independent variables. If only one of the coordinates, say x, is varied in the process of scanning the z plane while the other coordinate y is held fixed at discrete values, the above relations can be implemented by using only integrators (for the generation of

powers of x), summing amplifiers, and potentiometers. The operation in this case is rather laborious but requires only linear computing elements.

By contrast, instrumentation for the harmonic-synthesis procedure on the basis of polar coordinates requires multipliers or integrators to generate powers of r, and trigonometric resolvers or their equivalent to provide the harmonic terms $\sin k\theta$ and $\cos k\theta$. As will be discussed below, the harmonic-synthesis technique has the considerable advantage of producing not only the functions u and v, but also their partial derivatives with respect to r and θ simply by the addition of a few potentiometers and summing amplifiers.

5.11.5. Scan Methods. Among the scanning patterns to be considered here are linear, spiral, and circular paths in the z plane. In each case a certain scanning pattern is followed systematically while the functions u and v are observed.

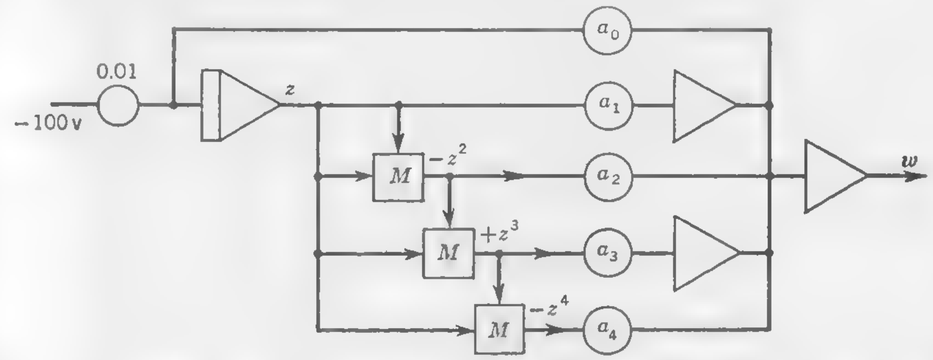


FIG. 5.11.4. Unscaled circuit for solving Eq. (5.11.28) for real roots with the aid of multipliers.

In *linear scanning* the x variable is allowed to change from negative to positive values while y is held fixed during each run of the computer. The variation of x may be accomplished by turning a number of multiplier potentiometers manually or through a computer servo, or by means of cascaded integrators. A particularly simple case is that of scanning the real axis, which arises if the equation is known to have real roots only or if complex roots are of no interest.

Suppose that it is desired to solve an equation of the form

$$w = a_4z^4 + a_3z^3 + a_2z^2 + a_1z + a_0 = 0 \tag{5.11.28}$$

It is possible to solve this equation directly by using multipliers to generate the powers involved and letting time be the independent variable z. This method is shown in Fig. 5.11.4. Each value of z, which makes $w = 0$, is a solution of the equation. The disadvantage of this open-loop method is that each multiplier will amplify the noise level in the circuit. It is possible to eliminate the multipliers by using integrators to obtain the required powers as shown in Fig. 5.11.5.

A third possible method for solving this equation consists in converting it to an nth-order differential equation, and solving by integration. Differentiating Eq. (5.11.28) four times, we obtain

$$\begin{aligned}
\frac{dw}{dz} &= 4a_4z^3 + 3a_3z^2 + 2a_2z + a_1 \\
\frac{d^2w}{dz^2} &= 12a_4z^2 + 6a_3z + 2a_2 \\
\frac{d^3w}{dz^3} &= 24a_4z + 6a_3 \\
\frac{d^4w}{dz^4} &= 24a_4
\end{aligned} \tag{5.11.29}$$

The above equations can be solved as shown in Fig. 5.11.6. Note that this latter method, which employs only a series of integrators, is the simplest to set up and adjust, since the coefficients in the equation appear only as initial conditions. The disadvantages of open-loop operation are also encountered in this case.

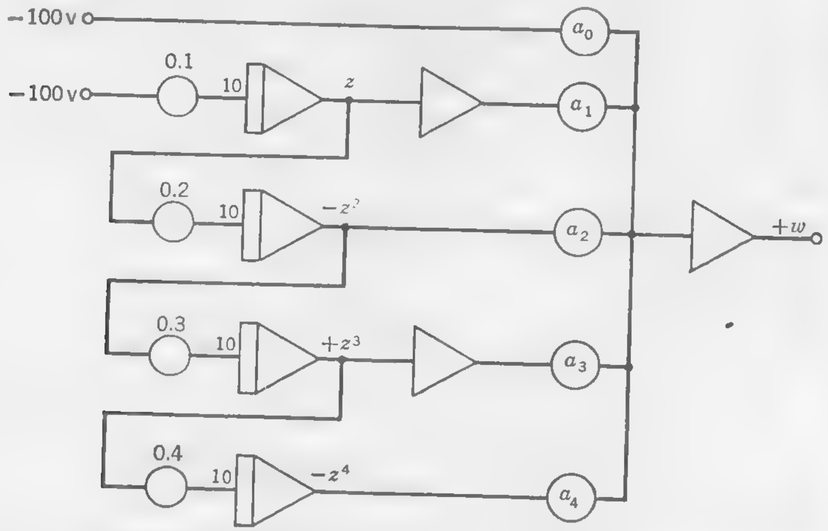


FIG. 5.11.5. Circuit for solving Eq. (5.11.28) for real roots without multipliers.

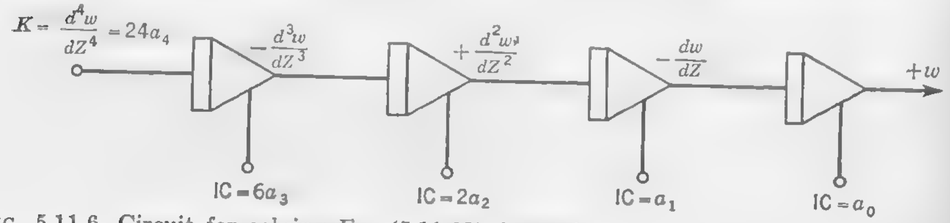


FIG. 5.11.6. Circuit for solving Eq. (5.11.28) for real roots with the aid of integrators.

Gephart[9] reduces the problem of finding the roots of polynomial equations to the equivalent problem of finding the eigenvalues of a matrix. The polynomial is first written as a determinant. For example, the polynomial equation of fourth degree

$$z^4 + b_3z^3 + b_2z^2 + b_1z + b_0 = 0 \tag{5.11.30}$$

may be expressed as

$$\begin{vmatrix} z & 0 & 0 & b_0 \\ -1 & z & 0 & b_1 \\ 0 & -1 & z & b_2 \\ 0 & 0 & -1 & z + b_3 \end{vmatrix} = 0 \tag{5.11.31}$$

Using the procedure previously described for eigenvalues, an augmented system of linear equations is set up on the computer, viz.,

$$\begin{aligned} T + t_1 &= k \\ T + zt_1 + b_0t_4 &= 0 \\ -t_1 + zt_2 + b_1t_4 &= 0 \\ -t_2 + zt_3 + b_2t_4 &= 0 \\ -t_3 + (z + b_3)t_4 &= 0 \end{aligned} \tag{5.11.32}$$

and the variable z is slowly varied to find those values for which T vanishes. Thus, if a polynomial with only real roots equivalent to a matrix with real eigenvalues is considered, all roots may be obtained in a single scanning process. The method can

be extended to complex roots, but it is less practical than the method of harmonic synthesis referred to and described later.

In general, roots located anywhere in the z plane are desired. For different settings of y_s a sequence of pairs of curves $u(x, y_s)$, $v(x,y_s)$ is drawn on an output table. Those values of x and y for which both u and v go through zero simultaneously are the roots of the polynomial. Since the equation was assumed to have real coefficients, all complex roots occur in conjugate pairs. Therefore, a scanning of the upper half of the z plane suffices to find all roots of the polynomial. In areas where u and v vanish approximately at the same argument values, the roots z_i can be found by interpolation, or the scan pattern is narrowed to determine x_i and y_i with the desired accuracy. The more general case of polynomials with complex coefficients, although not considered explicitly in this discussion, can be handled with relatively little additional programming effort, especially if Eqs. (5.11.24) and (5.11.25) are used to represent u and v.

The *spiral-scanning* technique described by Marshall[15] uses polar coordinates where r increases or decreases gradually while θ goes through a large number of revolutions. If the r variation is sufficiently slow compared with the θ variation, a dense coverage of the z plane is achieved. The method, originally developed for special-purpose computing equipment, is particularly attractive if the scan is performed at high computing speeds and exhibited on an oscilloscope screen. Roots located in the scan pattern are displayed on the screen by blips which are generated by coincidence signals from two gate circuits whenever the conditions $|u| < \epsilon_1$, $|v| < \epsilon_2$ are simultaneously satisfied. Although of limited accuracy, the method has the advantage of rapidly indicating the changes in the configuration of roots caused by parameter variation.

The *circular-scanning* technique employing the equations of harmonic synthesis (5.11.24) and (5.11.25) has been developed in various forms suitable for special-purpose as well as general-purpose computers. Circular paths described in the z plane give rise to variations in u and v that are observed by plotting these components against each other on an output table. The computer actually provides a conformal mapping of circles in the z plane into their equivalents in the w plane. The search for zeros of u and v and, hence, of w is conducted by a systematic procedure described below.

The method was originally used in the Isograph,[13,14] a high-precision mechanical instrument for polynomial solution, and has since been adapted to electronic analog computers.[15,16] It is based on a familiar principle from the theory of functions of complex variables, which states: A simple closed contour around the origin of the z plane which encloses m roots of the polynomial $w(z) = 0$ is mapped into a path encircling the origin of the w plane precisely m times. Thus, if a circle of radius r_s around the origin of the z plane is considered and the encirclements of the origin of the w plane by the mapped curve are counted, the number of roots z_i lying within the above circle, i.e., having a modulus $|z_i| < r_s$, may be found. This number increases by 1 for each root that is passed in the process of enlarging the radius r_s, and vice versa. Thus, by varying the radius r_s of the scan circle in the z plane the individual roots z_i can be systematically bracketed and their moduli $|z_i|$ determined by narrowing the brackets until the roots are isolated. The argument θ_i of each root is then determined by locating the point, or points, on the circle of radius r_i for which the image plot in the w plane passes through zero. The process is illustrated in Fig. 5.11.7, which shows several successive contour plots for decreasing scan radius in the case of a fourth-degree polynomial.

After acquiring some familiarity with the Isograph technique, the computer operator will be able to predict certain properties of the roots that are located within the scan circle. The character of the contour in the w plane reveals whether the roots are real or complex and whether they are located in the left or right half of the z plane.

Two methods of implementing the equations of harmonic synthesis on a general-purpose computer have been described by Bauer and Fifer.[16] For an nth-degree polynomial the equation solver requires a set of n resolvers or sine-cosine potentiometers geared in the ratio of 1: 2: 3: . . . n. Powers of r multiplied by the respective coefficients a_k are obtained by using suitably calibrated integrator circuits that are operated in the time interval between successive scan cycles. The alternate method employs n sine-cosine generators accurately adjusted to integral frequency ratios while the

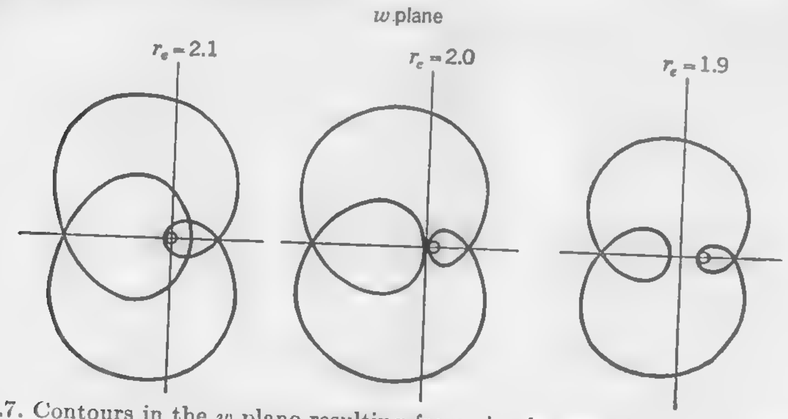


FIG. 5.11.7. Contours in the w plane resulting from circular scan for the polynomial $z^4 + 2.46z^3 - 1.46z^2 + 10.93z - 8 = 0$ having roots at $z = 1, -2, \sqrt{3} \pm i$. Contours in the vicinity of the root at $z = -2$ are shown.

powers of r are voltages obtained from multipliers and applied as initial values to the sine-cosine generators at the beginning of each scan operation.

NULL AND TRACK METHODS

5.11.6. Relaxation Method. In this method, one minimizes the modulus of the complex polynomial $w(z)$ by following a path in the xy plane composed of a sequence of line segments orthogonal with respect to each other, as illustrated in Fig. 5.11.3*e*. Starting at an arbitrary initial point z_0 a straight path is followed in a direction of decreasing values of the modulus until a relative minimum is reached. A new path perpendicular to the previous one is adopted and the process is continued, by following alternating path segments, until the condition $|w| = 0$ is attained, i.e., until the path has reached a root of the polynomial. The root to which this process converges depends upon the initial point chosen in the z plane. To find all roots of the polynomial the initial point must be changed a sufficient number of times.

A version of the relaxation technique applicable to the analog computer is described by Bubb.[17] In this formulation, the polynomial equation stated in Eq. (5.11.22) is represented as a vector sum by introducing polar coordinates. This yields

$$w(z) = a_n r^n e^{in\theta} + a_{n-1} r^{n-1(n-1)\theta} + \cdots + a_1 r e^{i\theta} + a_0 = N e^{i\Phi} \quad (5.11.33)$$

The term $Ne^{i\Phi}$ on the right is the resultant vector having the modulus N, which must be reduced to zero. Figure 5.11.8 illustrates the vector polygon representing a polynomial of fourth degree in the z plane.

On the computer the vectors are obtained in the following manner. A sine wave of fixed frequency ω is passed through n identical cascaded networks, each of which shifts the phase of the input signal by an angle θ and multiplies its magnitude by r. These networks may, therefore, be interpreted as complex operators multiplying their respective input signals by $z = re^{i\theta} = x + iy$. Each network requires basically one amplifier in parallel with an integrator. If the n intermediate voltages are added, the resulting output voltage of the computer is $Ne^{i\Phi}e^{i\omega t}$. This quantity equals zero only if $N = 0$ and, hence, according to Eq. (5.11.33) only if $x + iy$ is a root of the polynomial. N is measured as the amplitude of the sinusoidal output of the polynomial circuit. The value of N is reduced to zero by adjusting x and y, the gain factors of the amplifiers and integrators in the network, alternately in a minimizing sequence. Thus, if the process is started at $z_0 = x_0 + iy_0$, the coordinate x is varied, with y held constant, until a relative minimum of the amplitude N is reached. Next, x is held constant while y is varied, until a new relative minimum is reached, and this procedure is continued until N decreases to zero. Note that the phase angle Φ is not required and, hence, is not measured in the nulling process.

5.11.7. Method of Steepest Descent. The method of steepest descent differs from others by providing an *automatic* nulling process for finding the roots of the polynomial. The computer minimizes a related function in a continuous manner by following a path in the z plane along the gradient (see f in Fig. 5.11.3). This function must satisfy three requirements:

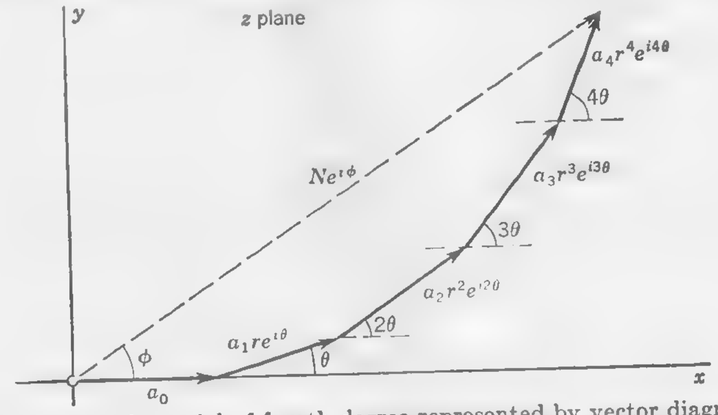


FIG. 5.11.8. Polynomial of fourth degree represented by vector diagram.

1. It must be nonnegative for all values of z.
2. It has minima only at points where $F = 0$.
3. These points must be the roots of the polynomial equation being considered.

Two simple functions satisfying these requirements are considered here.

$$F_1 = u^2 + v^2 = |w|^2 \quad (5.11.33)$$
$$F_2 = |u| + |v| \quad (5.11.34)$$

where u and v are given in Eqs. (5.11.24) and (5.11.25). Figure 5.11.9 illustrates the contour and gradient lines of F_1 for a typical fourth-order polynomial.

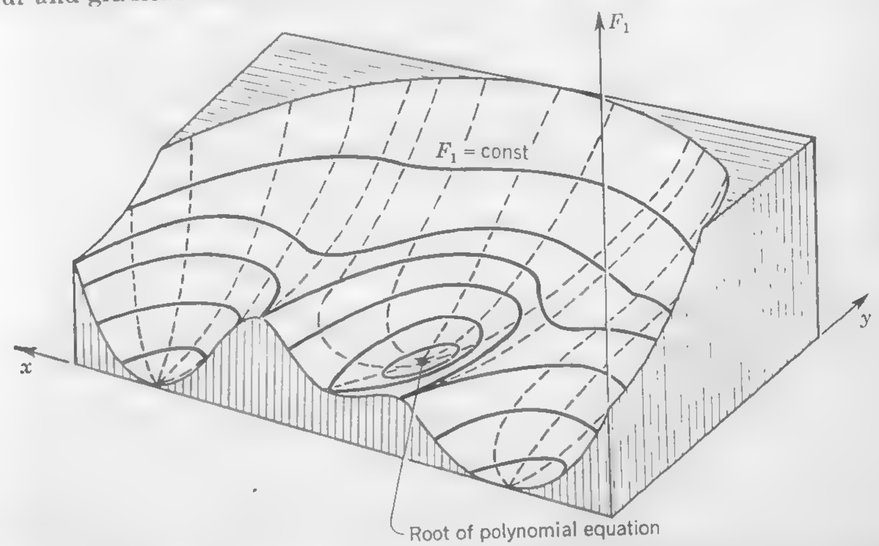


FIG. 5.11.9. Contours and curves of steepest descent of the function $F_1 = u^2 + v^2$ for a fourth-degree polynomial.

The basic relationship of the method of steepest descent expressed in polar coordinates, viz.,

$$\frac{1}{r}\frac{dr}{d\theta} = \frac{\partial F/\partial r}{(1/r)(\partial F/\partial \theta)} \quad (5.11.35)$$

ensures that r and θ are varied continuously in the direction prescribed by the gradient vector.

To satisfy this condition the computer solves two simultaneous first-order differential equations of the form

$$\frac{dr}{dt} = -cr^2 \frac{\partial F}{\partial r} \tag{5.11.36}$$

$$\frac{d\theta}{dt} = -c \frac{\partial F}{\partial \theta} \tag{5.11.37}$$

These equations are stable for all values of r and θ. The negative sign establishes a descending path, whereas c is an arbitrary constant determining the rate of descent. The partial derivatives of F may be expressed in terms of u, v and the partial derivatives of u. Required on the computer, therefore, are the quantities

$$\left.\begin{aligned} u &= \sum_{k=0}^{n} a_k r^k \cos k\theta \\ v &= \sum_{k=0}^{n} a_k r^k \sin k\theta \end{aligned}\right\} \tag{5.11.38}$$

$$r \frac{\partial u}{\partial r} = \sum_{k=1}^{n} k a_k r^k \cos k\theta \tag{5.11.39}$$

$$\frac{\partial u}{\partial \theta} = -\sum_{k=1}^{n} k a_k r^k \sin k\theta \tag{5.11.40}$$

The task of instrumenting these equations on the computer is simplified because of the structural similarity between u, v and the partial derivatives of u. It is worth noting that u and v are the terms required in the Isograph method. Thus, by a minor change in the computer operation, one may utilize the circuits designed for the steepest-descent method to obtain the additional information provided by the Isograph method, i.e., the determination of the number of roots in a closed region of the z plane.

A number of steepest-descent paths obtained by employing the function F_2 in the minimizing process are illustrated in Fig. 5.11.10 for the polynomial

$$w(z) = z^3 - 5z^2 + 8z - 6 = 0 \tag{5.11.41}$$

which has the roots

$$z_1 = 3 \qquad z_2 = 1 + i \qquad z_3 = 1 - i$$

The paths run in a generally diagonal direction through the network of u and v contours until they intersect one of the curves $u = 0$ or $v = 0$, which is then followed to the root.

5.11.8. Root-locus Tracking. Once the computer has acquired the root by the nulling process it may be operated in the automatic tracking mode. Thus, if one or several of the coefficients of the polynomial are varied, the steady-state condition will be altered, i.e., the computer will follow the root to its new location. This property is extremely useful to control-system engineers who use root loci to obtain information about the stability of servo systems.

If a closed-loop-system transfer function is expressed in the form

$$G(s) = \frac{K \dfrac{p(s)}{q(s)}}{1 + K \dfrac{p(s)}{q(s)}} \tag{5.11.42}$$

the root-locus plot determines the position of the zeros of the polynomial $q(s) + Kp(s)$ in the complex s plane as K is varied.

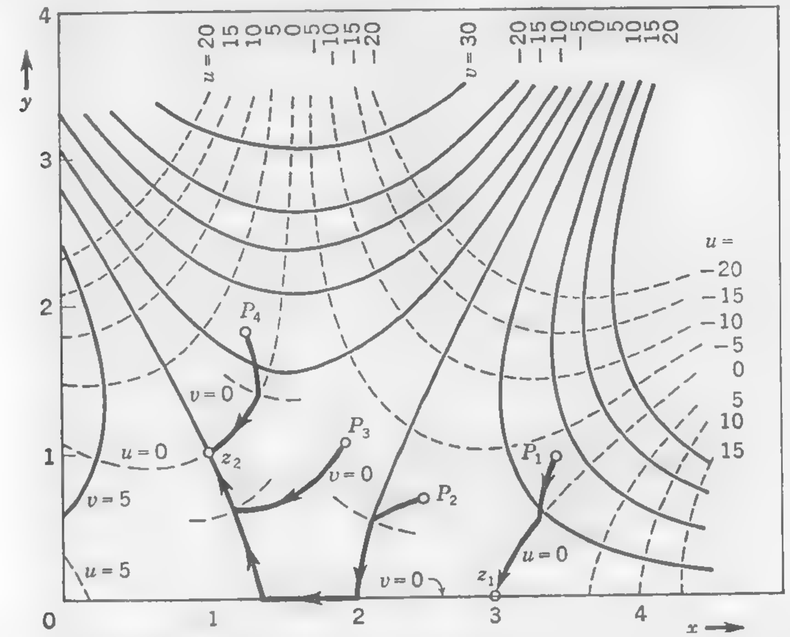


Fig. 5.11.10. Steepest-descent paths for a third-degree polynomial using the function $F_2 = |u| + |v|$. Roots are $z = 3$ and $1 \pm i$.

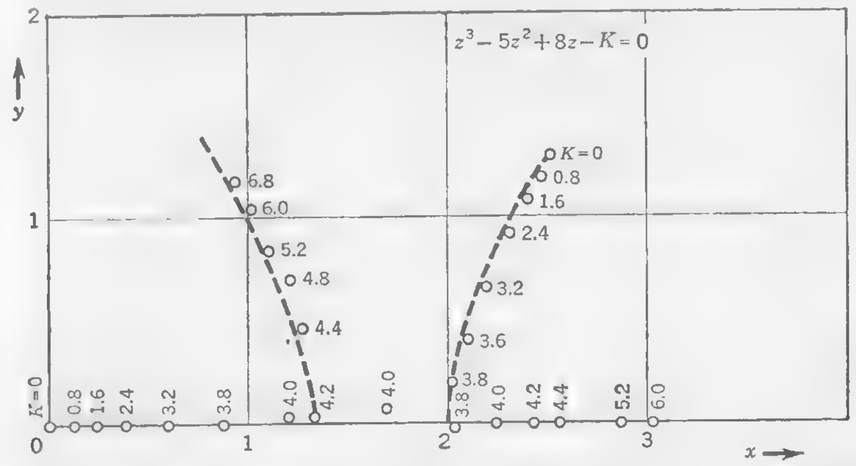


Fig. 5.11.11. Root locus obtained by the steepest-descent method for the polynomial

$z^3 - 5z^2 + 8z - K = 0$

- - - - Numerically computed root locus

○ Roots obtained by steepest descent

Figure 5.11.11 illustrates the root-locus plot obtained by varying the parameter K in the equation

$$w(z) = z^3 - 5z^2 + 8z - K = 0 \tag{5.11.43}$$

In control-system engineering, emphasis is often placed on root loci resulting from the variation of the loop gain K. In many instances, however, a parameter whose influence on stability must be determined affects one or more coefficients of $q(s)$ and $p(s)$. In the method of steepest descent, a root locus generated by these parameter

changes can be determined as readily as the conventional root locus resulting from a change in K.

5.11.9. Miscellaneous Methods. An nth-degree polynomial equation can be interpreted as the characteristic equation of an nth-order linear differential equation with constant coefficients. For example, with the fourth-degree polynomial equation

$$w = a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0 = 0 \tag{5.11.28}$$

one associates the differential equation

$$a^4 \frac{d^4\phi}{dt^4} + a_3 \frac{d^3\phi}{dt^3} + a_2 \frac{d^2\phi}{dt^2} + a_1 \frac{d\phi}{dt} + a_0\phi = 0 \tag{5.11.44}$$

Complex roots λ_k of the characteristic equation yield an oscillatory solution $\phi(t)$ containing the complex exponential terms $e^{\lambda_k t}$ where $\lambda_k = \delta_k + i\nu_k$. If the different

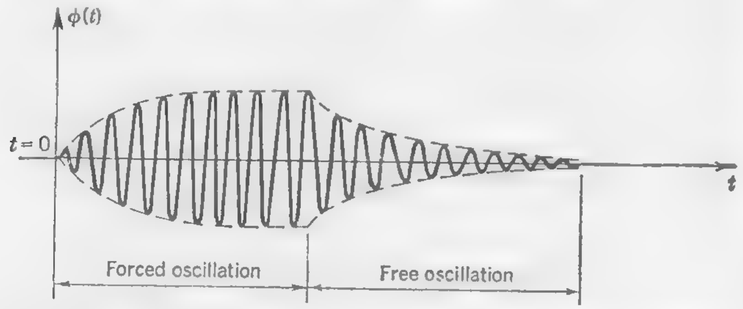


FIG. 5.11.12. Mode separation by means of sinusoidal excitation.

modes of oscillation are clearly recognized in the solution $\phi(t)$, the damping δ_k and the frequency ν_k may be determined from output plots of ϕ vs. t. In general, it will be necessary, however, to introduce sinusoidal forcing functions of various frequencies to provide excitation for the different modes of oscillation. Usually the critical frequencies are easily found in this manner by observing resonance peaks in the output variable $\phi(t)$. After removal of the excitation the particular oscillatory mode which has been excited to resonance persists in nearly pure form, and hence, its damping and frequency can be determined (Fig. 5.11.12).

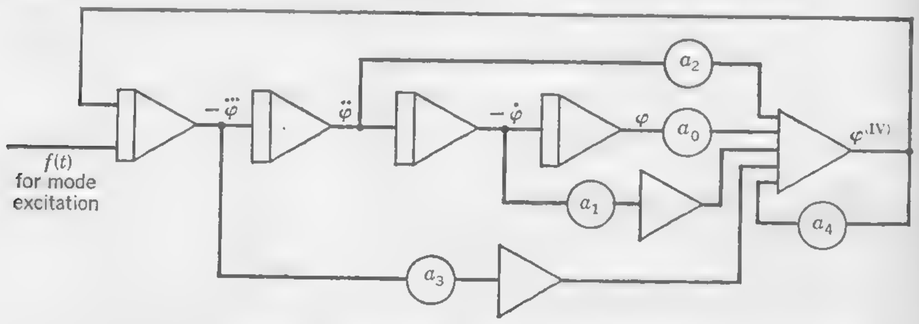


FIG. 5.11.13. Computer connection for finding the roots of Eq. (5.11.28).

The method is especially applicable in the case of Hurwitz polynomials, i.e., where the roots have a negative real part since these lead to damped oscillations. In the case of roots with positive real parts, a reversal of the time parameter is required in order to obtain a sufficiently long-sustained oscillation without overloading the computer.

The computing equipment required for this method is shown in Fig. 5.11.13. It can be seen that this method is less elaborate and also faster than those previously considered. However, it is not as accurate owing to the need for reading δ_k and ν_k from a graph and to the difficulty of obtaining pure mode shapes. The method is also well

suited for the purpose of mode tracing and observation of the effect of parameter variations upon the stability of the linear oscillatory system represented by the polynomial equation.

It should be noted that there is a simple technique for finding roots whose absolute value is so large as to exceed the voltage range of the computer. Ordinarily this situation would be remedied by a scale change in the variables x, y and u, v. This can be avoided by introducing the new variable $\zeta = 1/z$ which transforms the original polynomial Eq. (5.11.28) into

$$P(\zeta) = a_0\zeta^4 + a_1\zeta^3 + a_2\zeta^2 + a_3\zeta + a_4 = 0 \tag{5.11.45}$$

having the same coefficients as the original equation but in reverse order. Since for a root z_i with large modulus $|z_i| > 1$ the corresponding root $\zeta_i = 1/z_i$ falls within the unit circle, the entire set of roots of Eq. (5.11.28) can be evaluated, in general, without a scale change, by solving Eqs. (5.11.28) or (5.11.45) alternately, depending on the moduli in question.

There is a simple technique for checking the proper operation of a polynomial solver which is applicable to different types. If all the coefficients are set to unity, then the roots of the polynomial equation lie on the unit circle and are symmetrical about the real axis. This is easily shown by multiplying the polynomial equation

$$z^n + z^{n-1} + \cdots + z^2 + z + 1 = 0 \tag{5.11.46}$$

by the factor $(z - 1)$. This produces

$$(z - 1)(z^n + z^{n-1} + \cdots + z^2 + z + 1) = 0 \tag{5.11.47}$$

or

$$z^{n+1} = 1 \tag{5.11.48}$$

The roots of this last equation, however, are the $n + 1$ roots of unity. Hence, subtracting the root $z = 1$ from this set leaves the roots of Eq. (46).

LINEAR PROGRAMMING

5.11.10. Theory. A *linear-programming problem* requires one to maximize the *objective function* of n variables,

$$Z = \sum_{k=1}^{n} \frac{x_k}{\alpha_k} \qquad (x_k \geq 0) \tag{5.11.49}$$

subject to the m restrictions (constraints)

$$\begin{aligned} &\sum_{k=1}^{n} \frac{x_k}{a_{1k}} \leq 1 \\ &\sum_{k=1}^{n} \frac{x_k}{a_{2k}} \leq 1 \\ &\cdots\cdots\cdots \\ &\sum_{k=1}^{n} \frac{x_k}{a_{mk}} \leq 1 \end{aligned} \tag{5.11.50}$$

where α_k and a_{ik} are given constants. In general, this problem can be solved with $(n + 2m + 2p)$ amplifiers[19] where n = number of variables, m = number of restrictions, p = number of negative quantities among the a_{ik}.

The variables $x_1, x_2, \ldots, x_n$ can be treated as coordinates of a point in n-dimensional Euclidean space. The point with these coordinates will be called the *objective point*. The objective function (5.11.49) is continuous and single-valued everywhere within this space and, therefore, can be used to define a gradient vector, grad Z. Then it is possible to define a vector

$$\mathbf{f} = K \operatorname{grad} Z + \sum_{j=1}^{m} \delta_j \mathbf{N}_j \tag{5.11.51}$$

where K = constant,

$$\mathbf{N}_j = -\left(\frac{\mathbf{i}_1}{a_{j1}} + \frac{\mathbf{i}_2}{a_{j2}} + \cdots + \frac{\mathbf{i}_n}{a_{jn}}\right)$$

and the m quantities $\delta_1, \delta_2, \ldots, \delta_m$, one for each inequality, are such that

$$\begin{aligned} \delta_j &= 0 \qquad \text{when } \sum_{k=1}^{n} \frac{x_k}{a_{jk}} \leq 1 \\ \delta_j &= 1 \qquad \text{otherwise} \end{aligned} \tag{5.11.52}$$

The kth component of f is

$$f_k = \frac{K}{\alpha_k} - \sum_{j=1}^{m} \frac{\delta_j}{a_{jk}} \tag{5.11.53}$$

and this component, like the vector f itself, depends on the coordinates of the objective point because of the presence of the δ_j.

Let the position of the objective point be described by the vector

$$\mathbf{r} = x_1\mathbf{i}_1 + x_2\mathbf{i}_2 + \cdots + x_n\mathbf{i}_n$$

and define a constant scalar γ such that

$$\frac{d\mathbf{r}}{dt} = \gamma \mathbf{f} \tag{5.11.54}$$

Equation (5.11.54) describes the motion of the objective point along the gradient of Z as the solution is approached. The solution is to be the value which r finally attains. The kth component in Eq. (5.11.54) is

$$\frac{dx_k}{dt} = \gamma\left(\frac{K}{\alpha_k} - \sum_{j=1}^{m} \frac{\delta_j}{a_{jk}}\right) \tag{5.11.55}$$

The objective point moves through the allowed region with a velocity γK grad Z, until it reaches a restriction such as the jth equation (5.11.50). The actual rebound is very small, but is shown greatly exaggerated in Fig. 5.11.14.

To prevent break-through into the restricted region, we must have

$$-K \operatorname{grad} Z \cdot \mathbf{N}_j < \mathbf{N}_j^2 \tag{5.11.56}$$

or the equivalent

$$K < \frac{\sum_{k=1}^{n} \frac{1}{a_{jk}^2}}{\sum_{k=1}^{n} \frac{1}{a_{jk}\alpha_k}} \tag{5.11.57}$$

which must be satisfied for every j.

5.11.11. Analog-computer Solution. An analog-computer setup to solve the linear-programming systems comprises three types of computing elements (Fig. 5.11.15): (1) n summing integrators limited to allow positive outputs only, (2) m summing amplifiers, one for each restriction, (3) m switches, one for each restriction. The switch is shown in Fig. 5.11.15c. Its properties are described as follows:

$$\begin{aligned} e_{out} &= 0 \text{ volt} \qquad \text{when } e_{in} > 0 \\ e_{out} &= E \text{ volts} \qquad \text{when } e_{in} < 0 \end{aligned} \tag{5.11.58}$$

Figure 5.11.16 shows a complete computer setup.

In commercial linear-programming computers a matrix of potentiometers is provided to set the constants $1/a_{jk}$ and $1/\alpha_k$. If the programming problem is nonlinear,

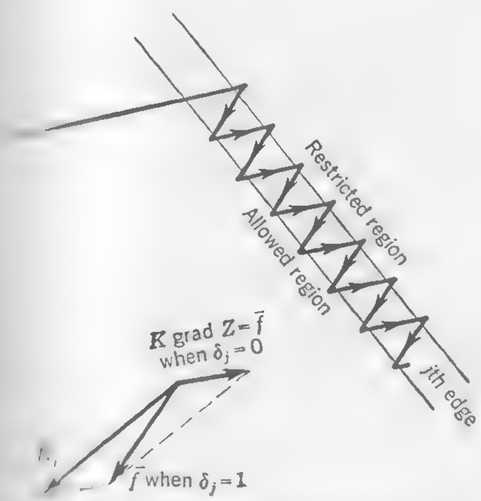


FIG. 5.11.14. Motion of objective point in neighborhood of a restriction.

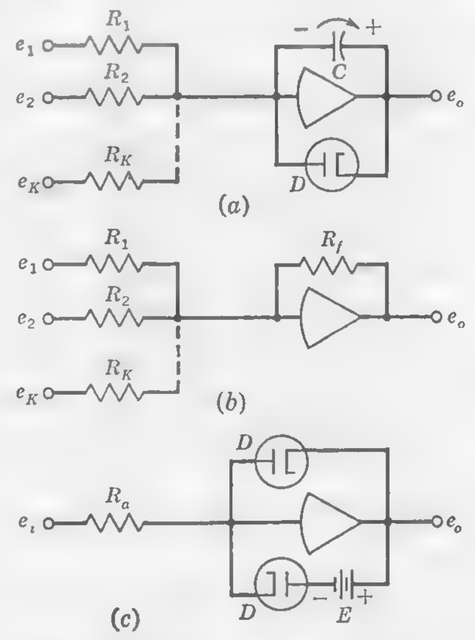


FIG. 5.11.15. The three basic elements for linear programming.

even it is necessary to use the usual auxiliary equipment in the form of function generators in conjunction with this computer. Because of the nature of the computer it is also readily adapted to solving simultaneous linear algebraic equations, inverting matrices, and evaluating determinants.

SPECIAL TECHNIQUES FOR PARAMETER OPTIMIZATION

5.11.12. Parameter Optimization. A frequent requirement in systems analysis is to find a set of system parameters $a, b, c, \ldots$ which will maximize (or minimize) a specified function $h = h(a,b,c, \ldots)$ serving as a measure of system effectiveness. Such a function is variously called response function, optimization criterion, objective function, payoff function, etc.[20] In the analysis of dynamic systems the response function usually is a functional derived from the solution $y_i(t;a,b,c, \ldots)$, $(i = 1, 2, \ldots, n)$ of a set of differential equations containing the system parameters $a, b, c, \ldots$ (*example:* hit probability of a missile, mean-square error of a variable, radius of action of an aircraft). In the following it will be assumed that a preliminary analysis has narrowed the parameter ranges so that there is only one maximum (or minimum) in the region of interest of the parameter space.

The task of parameter optimization involves computation of the response function h and a variation of parameters $a, b, c, \ldots$ until a maximum (or minimum) of h is obtained. The nulling, tracking, and scanning methods described in Arts. 5.11.4 to 5.11.11 may apply if the computation of h does not involve the solution of differential equations. In the latter case stepwise parameter variations are necessary. Analog computers with automatic programming features (Sec. 4.2) will aid in sequential computations; the search for optimum parameters will be speeded up if a sufficiently accurate repetitive analog computer is available. To obtain a sequence of parameter values which converges to the maximum (or minimum), one may start with initial parameter values $a = a_0$, $b = b_0$, $c = c_0$, . . . and

1. Vary $a, b, c, \ldots$ to compute finite-difference approximations to the gradient components $\partial h/\partial a, \partial h/\partial b, \ldots$. The next parameter combination $a = a_1$, $b = b_1$, $c = c_1$, . . . is then taken at some distance from the initial choice and in a direction corresponding to the steepest ascent or descent, i.e., by making all parameter increments proportional to the respective gradient components, and using the same constant of proportionality (*basic gradient method*).[21,22,23]

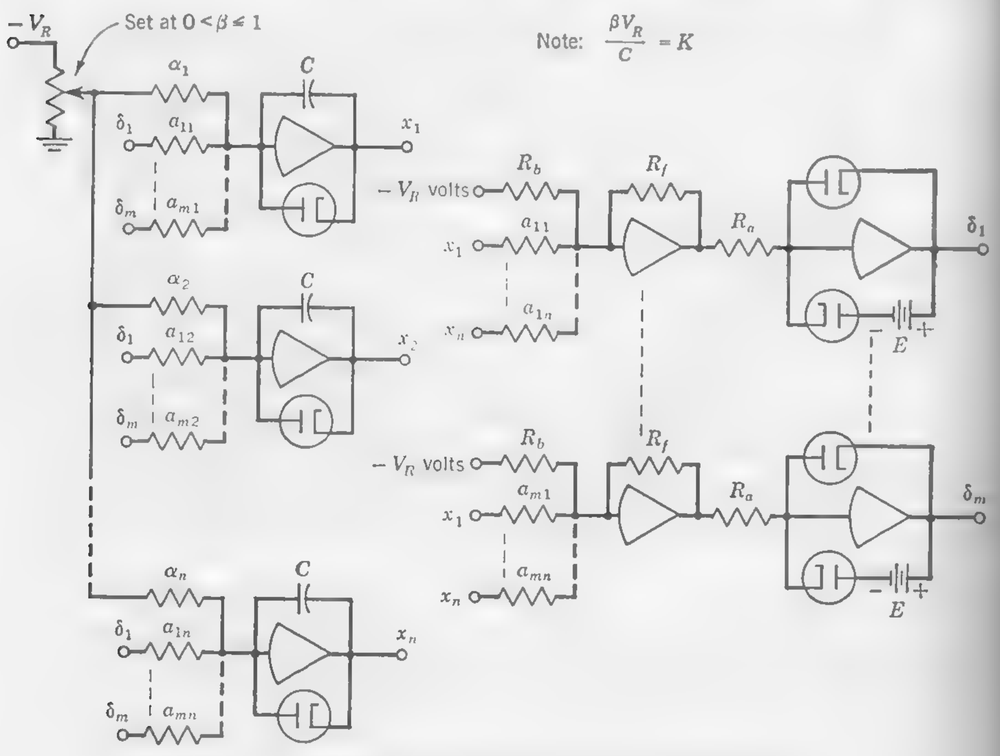


Fig. 5.11.16. Computer diagram for linear programming.

2. Compute the gradient components $\partial h/\partial a, \partial h/\partial b, \ldots$ as the solution of a set of differential equations programmed on the computer (*method of parameter influences*, Art. 5.11.13), and proceed as above.

3. Vary the parameters $a, b, c, \ldots$ one at a time until a maximum or minimum is reached (*relaxation method*). A useful variation of this method is to compute $h(a,b,c, \ldots)$ for three or more values of the parameter varied, to find the maximum of an interpolation parabola through these points, then to proceed to the next parameter, etc.

4. Vary the parameters systematically (*scanning*) or at random (*random search*) while a maximum-storing integrator (Fig. 5.11.17) temporarily retains the largest value of $h(a,b,c, \ldots)$ obtained, and similarly in the case of a minimizing sequence.[24,25,26]

The underlying principle of programming the computer for sequential operation to provide optimizing sequences of parameter values is closely related to the concepts used in self-adaptive control systems.[27] The technique permits the human computer operator to reduce the effort of trial-and-error search and to transfer certain time-consuming routines entirely to the computer. The technique should, however, not be viewed as a means of replacing human intelligence in systems analysis.

5.11.13. The Method of Parameter Influences. The gradient components $\partial h/\partial a, \partial h/\partial b, \ldots$ needed for optimization of dynamic systems by the gradient method are related to the partial derivatives $\partial y_i/\partial a, \partial y_i/\partial b, \ldots$ with respect to $a, b, \ldots$ associated with the solution $y_i = y_i(t;a,b, \ldots)$ of the system differential equations. These partial derivatives will be referred to as *parameter-influence coefficients*.[27]

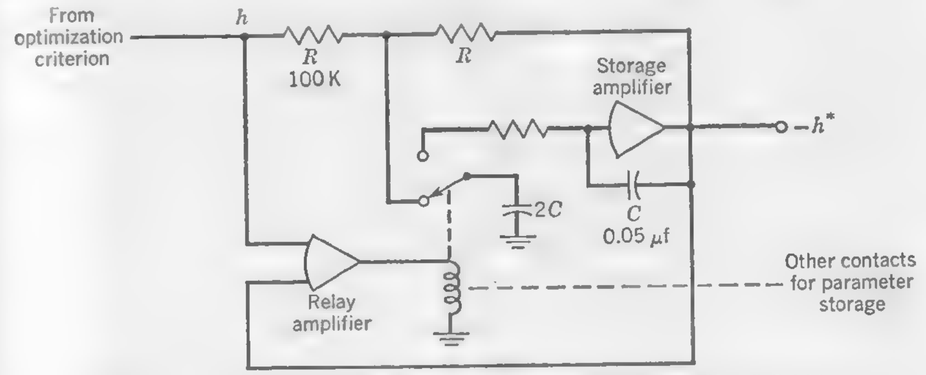


Fig. 5.11.17. Switching and storage circuits (see also Fig. 3.3.11).

Consider, for example, the problem of matching the output $y(t;a,b)$ of a dynamic system with two unspecified parameters a and b against an input function $y_1(t)$, in terms of least-square errors. Thus, the optimization criterion is given by

$$h(a,b) = \int_0^T [y(t;a,b) - y_1(t)]^2 \, dt \tag{5.11.59}$$

and the gradient components are

$$\frac{\partial h}{\partial a} = 2\int_0^T [y(t;a,b) - y_1(t)] \frac{\partial y}{\partial a} \, dt$$
$$\frac{\partial h}{\partial b} = 2\int_0^T [y(t;a,b) - y_1(t)] \frac{\partial y}{\partial b} \, dt \tag{5.11.60}$$

To find the parameter-influence coefficients $\partial y/\partial a, \partial y/\partial b$ needed in Eqs. (60) as functions of time, new differential equations are formed from the system equation, say

$$\frac{d^2y}{dt^2} + a\frac{dy}{dt} + by = y_1(t) \tag{5.11.61}$$

by differentiation with respect to the parameters a and b, thus

$$\frac{d^2}{dt^2}\left(\frac{\partial y}{\partial a}\right) + a\frac{d}{dt}\left(\frac{\partial y}{\partial a}\right) + b\frac{\partial y}{\partial a} = -\frac{dy}{dt} \tag{5.11.62a}$$

and

$$\frac{d^2}{dt^2}\left(\frac{\partial y}{\partial b}\right) + a\frac{d}{dt}\left(\frac{\partial y}{\partial b}\right) + b\frac{\partial y}{\partial b} = -y \tag{5.11.62b}$$

The solution of Eqs. (5.11.62), which may be performed on the computer concurrently with the solution of Eq. (5.11.61), yields the desired influence coefficients.[27] Note that in the case of time-invariant linear systems the subsidiary Eqs. (5.11.62) contain the

same homogeneous part as the primary Eq. (5.11.61) and use the output variable $y(t)$ and its time derivative as driving functions. Miller and Murray[28] have formulated equations of the type (62) in connection with error analysis of computer solutions and refer to them as *sensitivity equations*. The method applies to more general problems within the limitations of the computing equipment available.

References

1. Soroka, W. W.: *Analog Methods in Computation and Simulation*, McGraw-Hill Book Company, Inc., New York, 1954.
2. Korn, Granino A., and T. M. Korn: *Electronic Analog Computers*, McGraw-Hill Book Company, Inc., New York, 1956.
3. Murray, F. J.: Linear Equation Solvers, *Quart. Appl. Math.*, **7**:263–274 (1949).
4. McCool, W. A.: D-C Analog Solution of Simultaneous Algebraic Equations, *NRL Rept.* 3533, Sept. 9, 1949.
5. Korn, G. A.: Stabilization of Simultaneous Equation Solvers, *Proc. IRE*, **37**:1000–1002 (1949).
6. Adcock, W. A.: An Automatic Simultaneous Equation Computer and Its Use in Solving Secular Equations, *Rev. Sci. Instr.*, **19**:181–187 (1948).
7. Goldberg, E. A., and G. W. Brown: An Electronic Simultaneous Equation Solver, *J. Appl. Phys.*, **19**:339–345 (1948).
8. Walker, R. M.: An Analogue Computer for the Solution of Linear Simultaneous Equations, *Proc. IRE*, **37**:1467–1473 (1949).
9. Gephart, F. L.: Linear Algebraic Systems and the REAC, *Mathematical Tables and Other Aids to Computation*, **6**:190–203 (1952).
10. Murray, F. J.: *The Theory of Mathematical Machines*, King's Crown Press, New York, 1947.
11. Wilbur, J. B.: The Mechanical Solution of Simultaneous Equations, *J. Franklin Inst.*, **222**:715–724 (December, 1936).
12. Roberson, R. E., and W. A. McCool: A Note on the Analog Solution of Determinants, *NRL Rept.* 3557, Oct. 18, 1949.
13. Dietzold, R. L.: The Isograph—A Mechanical Root-Finder, *Bell Labs. Record*, **16**:130–134 (1937–1938).
14. Brown, S. L., and L. L. Wheeler: A Mechanical Method for Graphical Solution of Polynomials, *J. Franklin Inst.*, **231**:223–243 (1941).
15. Marshall, B. O., Jr.: The Electronic Isograph for Roots of Polynomials, *J. Appl. Phys.*, **21**:307–312 (1950).
16. Bauer, L., and S. Fifer: The Solution of Polynomial Equations on the REAC, *Project Cyclone Symposium I*, Reeves Instrument Corp., New York, pp. 31–36, 1951.
17. Bubb, F. W.: A Circuit for Generating Polynomials and Finding Their Zeros, *Proc. IRE*, **39**:1556–1561 (1951).
18. Levine, L., and H. F. Meissinger: An Automatic Analog Computer Method for Solving Polynomials and Finding Root Loci, *IRE Natl. Convention Record*, part IV, pp. 146–172, 1957.
19. Pyne, I. B.: Linear Programming on an Analogue Computer, *Communications and Electronics*, **24**:139–143 (May, 1956).
20. Goode, H. H., and R. E. Machol: *System Engineering*, McGraw-Hill Book Company, Inc., New York, 1957.
21. Courant, R.: Variational Methods for the Solution of Problems of Equilibrium and Vibration, *Bull. Am. Math. Soc.*, **49**:1–23 (1943).
22. Box, G. E. P., and K. B. Wilson: On the Experimental Attainment of Optimum Conditions, *J. Roy. Statis. Soc.*, B, **13**:1 (1951).
23. Tompkins, C. B.: Methods of Steep Descent, in *Modern Mathematics for the Engineer*, pp. 448–479, E. F. Beckenbach (ed.), McGraw-Hill Book Company, Inc., New York, 1956.
24. Davies, O. L.: *Design and Analysis of Industrial Experiments*, chap. 11, Hafner Publishing Company, New York, 1956.
25. Munson, J. K., and A. I. Rubin: Optimization by Random Search on the Analog Computer, *Trans. IRE/PGEC*, **8**:200, 1959.
26. Favreau, R. R., and R. Franks: Random Optimization by Analog Techniques, *Proc. Second International Conference for Analog Computation*, Strasbourg, September, 1958.
27. Margolis, M., and C. T. Leondes: A Parameter Tracking Servo for Adaptive Control Systems, *IRE Convention Record*, WESCON 1959, part 4, pp. 104–115, August, 1959.
28. Miller, K. S., and F. J. Murray: The Mathematical Basis for the Error Analysis of Differential Analyzers, *J. Math. Phys.*, nos. 2 and 3, 1953.

PART 12: ECONOMIC-DISPATCH COMPUTERS FOR POWER SYSTEMS

By R. P. Washburn

5.12.1. Introduction. A power-system load dispatcher must maintain his electric production equal to his load, maintain system frequency on isolated systems, or maintain specified power flows on tie lines to other utilities; optimum scheduling will result in minimum production cost per hour or per day. *Economic-dispatch computers* (EDC's) can calculate optimum scheduling of power output for each power source; some economic-dispatch computers incorporate actual control functions.

Estimated annual fuel savings obtainable with EDC's may approach $50 per megawatt installed. For a typical 1,000-megawatt system this saving can amount to $40,000 to $50,000 per year and thus may warrant an investment of $250,000. Over three-quarters of these savings could, however, also be realized through the uses of tabulated *loading schedules* obtained on a digital computer and up-dated for each system change.

5.12.2. Problem Statement. The objective will be to find generator outputs $P_1, P_2, \ldots, P_n$ which minimize the *total cost* (dollars per hour)

$$C = \sum_{k=1}^{n} F_k(P_k) \tag{5.12.1}$$

where $F_k(P_k)$ represents the cost of power generation as a function of generated output; the function (1) must be minimized subject to the constraint of a given system *power demand*

$$P_D = P_T - P_L = \sum_{k=1}^{n} P_k - P_L(P_1, P_2, \ldots, P_n) \tag{5.12.2}$$

where P_L is the power lost in the transmission networks. It is customary to approximate the *transmission loss* P_L in the form

$$P_L = \sum_{i=1}^{n} \sum_{k=1}^{n} B_{ik} P_i P_k \qquad (B_{ik} = B_{ki}) \tag{5.12.3}$$

where the B_{ik} are constant *transmission-loss coefficients* determined by the transmission networks linking loads and generators.

5.12.3. Optimum Scheduling with Negligible Transmission Losses. If transmission losses can be neglected ($P_L = 0$), the total cost (1) will be minimized if

$$\frac{dF_1}{dP_1} = \frac{dF_2}{dP_2} = \cdots = \frac{dF_n}{dP_n} = \lambda \tag{5.12.4}$$

the *incremental production costs* dF_k/dP_k (dollars per megawatt hour) of all generators must be equal to one another and equal the incremental cost λ at a hypothetical load center. In practice, each dF_k/dP_k is given in the form of a curve plotted against P_k. A set of such curves (Fig. 5.12.1*a*) yields $P_1, P_2, \ldots, P_n$ for each value of λ, so that the desired value of $P_T = P_1 + P_2 + \cdots + P_n$ can be matched by trial and error. The process can be mechanized by a *loading slide rule* having movable scales for each dF_k/dP_k, or by a simple electronic "slide-rule computer" (Fig. 5.12.1*b*) which computes generator outputs, station outputs, and total power for various values of λ. The machine shown provides for six generators located in two steam stations. A *net-interchange setter* is also provided to account for tie-line net interchanges. Each station has its *fuel-cost setter* to set in the cost of fuel at that station. The amplifier changes the

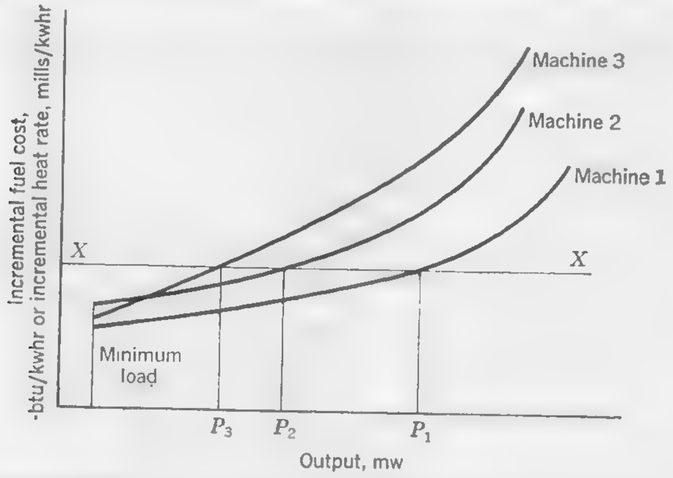


FIG. 5.12.1*a*. Incremental-fuel-rate curves (incremental-heat-rate curves) for three machines, showing machine outputs for a given incremental production cost.

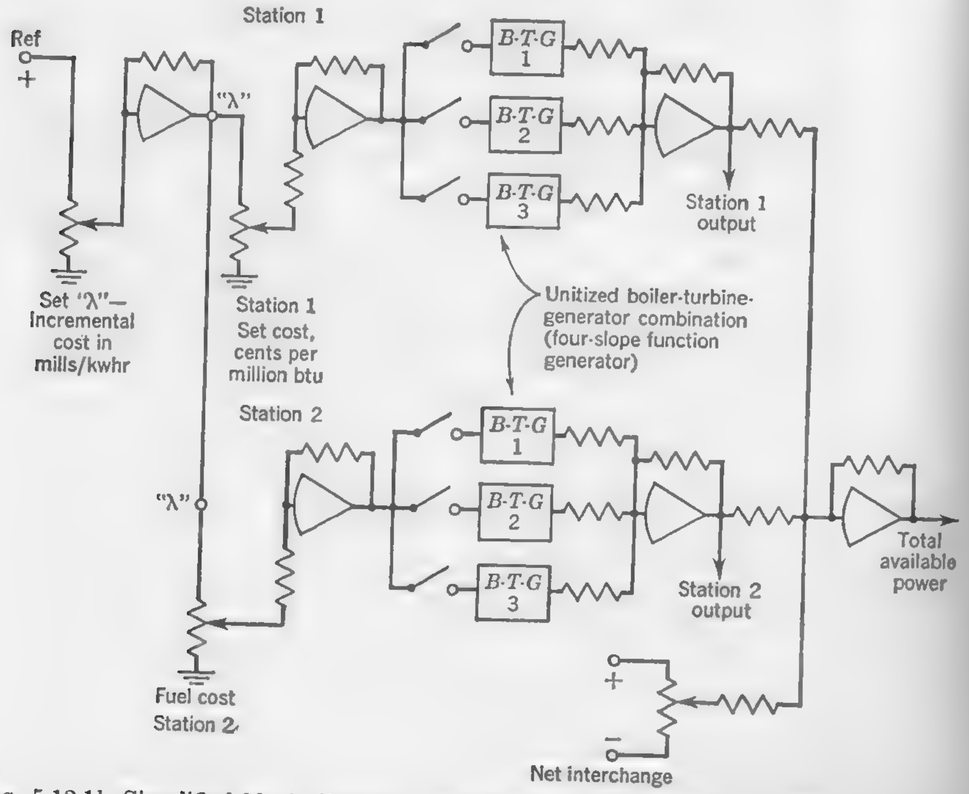


FIG. 5.12.1*b*. Simplified block diagram of an electronic "slide-rule" computer for economic loading of six generators in two stations with a net-interchange setter.

scaling so that the function-generator input is proportional to heat input into the boiler in btu $\times 10^6$ per hr. The function generator converts heat input to optimum generator output in megawatts for each unit. These are added for a station output; the sum of station outputs and net interchange yields the total power available for the preset value of λ. The machine does *not* adjust itself for the specified power demand P_D.

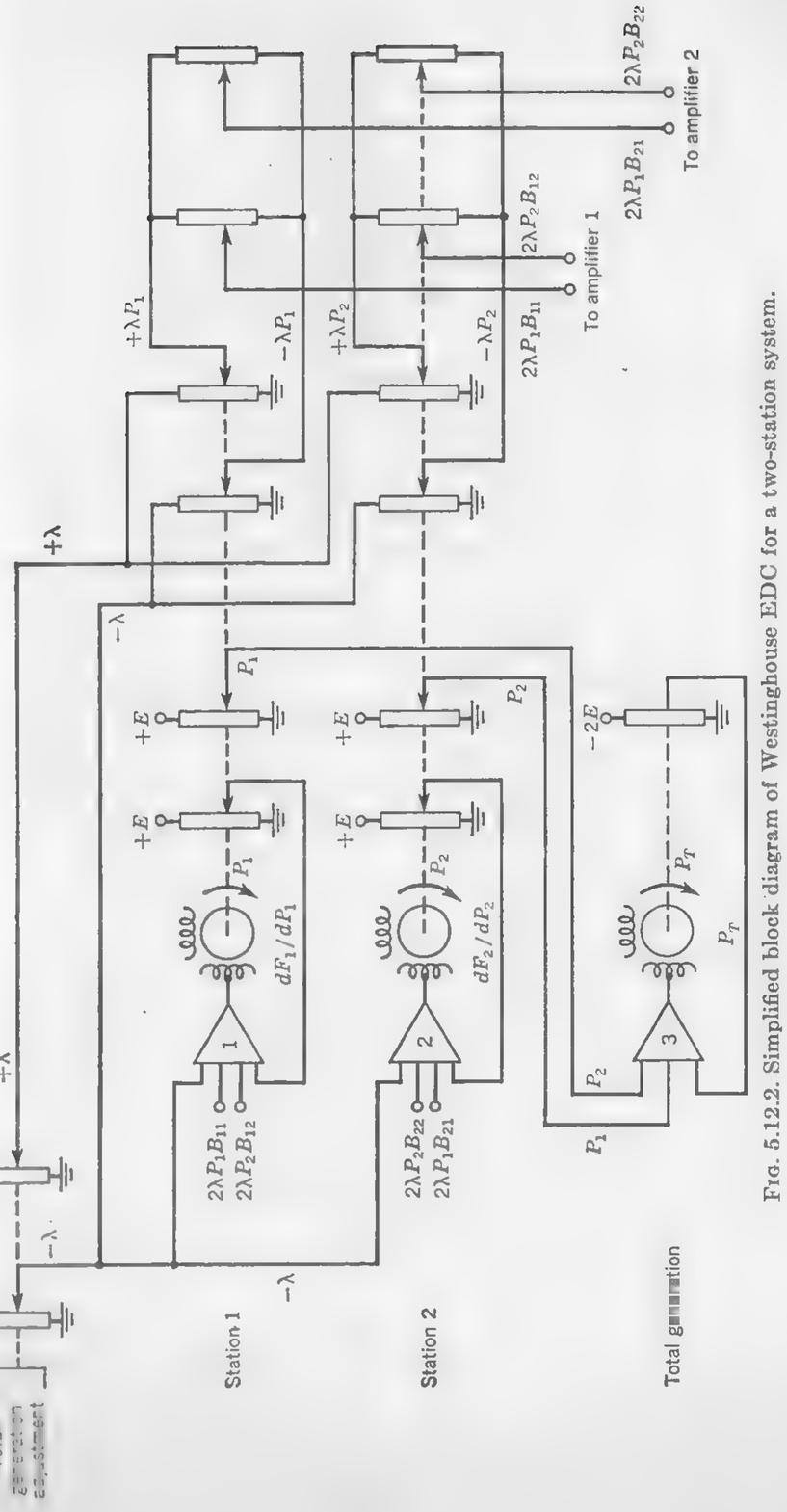


FIG. 5.12.2. Simplified block diagram of Westinghouse EDC for a two-station system.

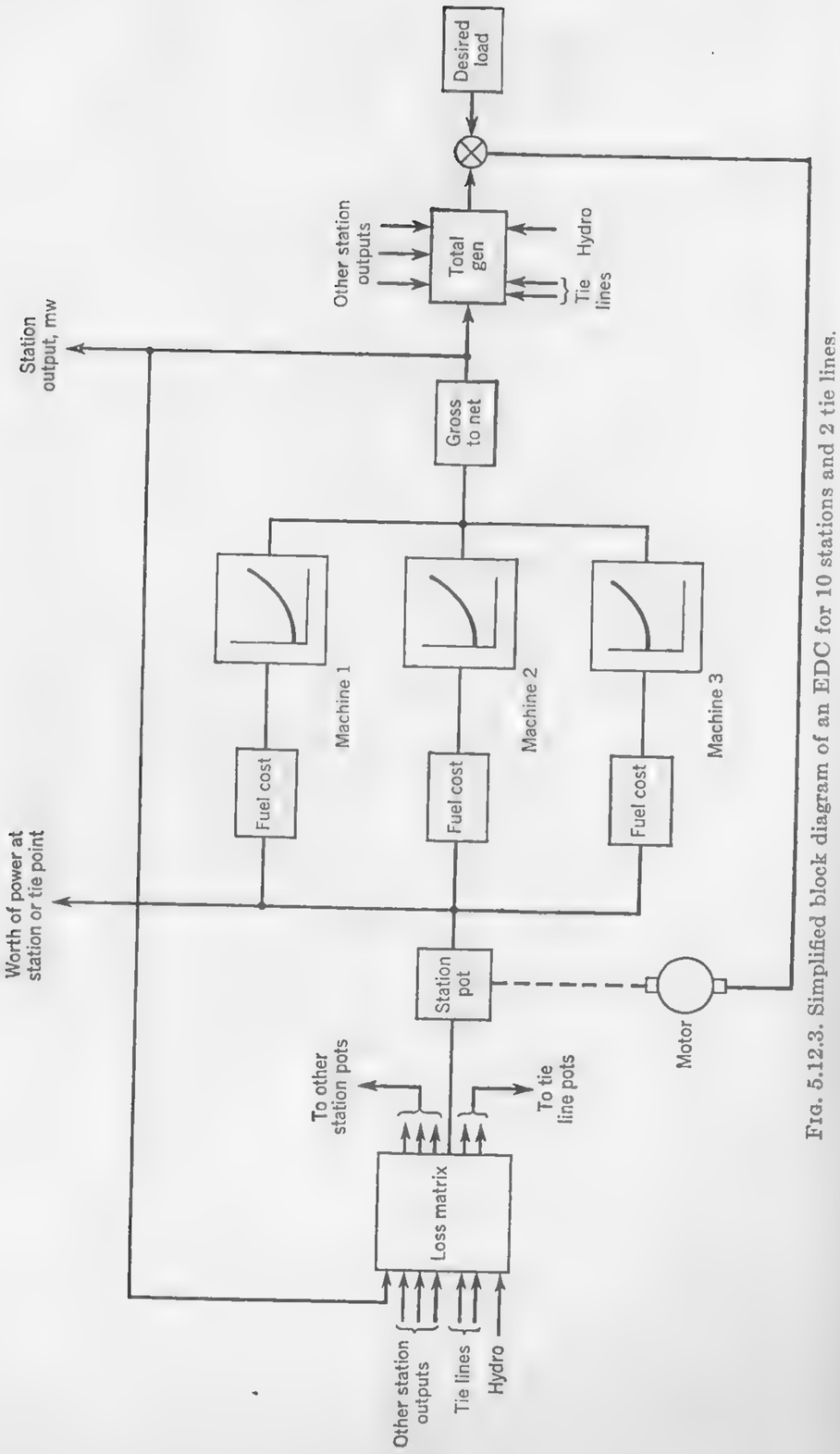


Fig. 5.12.3. Simplified block diagram of an EDC for 10 stations and 2 tie lines.

5.12.4. Complete Economic-dispatch Computers. If transmission losses cannot be neglected, the optimalizing condition becomes

$$\lambda_1(P_1,P_2, \ldots ,P_n) = \lambda_2(P_1,P_2, \ldots ,P_n) = \cdots = \lambda_n(P_1,P_2, \ldots ,P_n) = \lambda \tag{5.12.5a}$$

where λ is a constant (Lagrange multiplier) determined by the power demand, and where

$$\lambda_i(P_1,P_2, \ldots ,P_n) = \frac{dF_i}{dP_i}\frac{1}{1-(\partial P_L/\partial P_i)} = \frac{dF_i}{dP_i}\frac{1}{1-2\sum_{k=1}^{n} B_{ik}P_k} \qquad (i = 1, 2, \ldots, n) \tag{5.12.5b}$$

measures the incremental cost of power delivered from the ith generator to the hypothetical load center, including the effect of transmission losses; $(1 - \partial P_L/\partial P_i)^{-1}$ is called the transmission *penalty factor* for the ith generator. For optimum operation, the incremental costs are again equal for all generators.

The *Westinghouse EDC* (Fig. 5.12.2) enforces the relations (5) in the forms

$$e_i = \frac{dF_i}{dP_i} + 2\lambda\sum_{k=1}^{n} B_{ik}P_k - \lambda = 0 \qquad (i = 1, 2, \ldots, n) \tag{5.12.6}$$

through n servos driven by the respective "errors" ϵ_i; dF_i/dP_i is produced by a function potentiometer. $P_1, P_2, \ldots, P_n$ and $P_T = P_1 + P_2 + \cdots + P_n$ are read out, and λ is varied until the desired power output is obtained.

The all-solid-state *General Electric EDC* is of the type illustrated in Fig. 5.12.3, except that the actual computer involves four separate area EDC's and a complex servo system yielding optimum interarea interchanges of power. In Fig. 5.12.3, each dF_i/dP_i is generated as a function of P_i by a diode function generator and a fuel-cost-setting potentiometer. $\sum_{k=1}^{n} B_{ik}P_k$ is obtained from a *loss matrix* of summing amplifiers, phase inverters, and potentiometers, and a single servo performs n multiplications by λ. The machine yields generator outputs, station outputs, and cost of power for each station and has digital read-out.

5.12.5. EDC's as Control Systems.[8,11,12] **Digital EDC's.** Automatic-control versions of EDC's feature continuous on-line control of generator outputs according to economic criteria modified by practical operating considerations. Various semiautomatic systems are also in use.

Because of the relatively slow computing speeds and large information storage (3,000-potentiometer loss matrices for large-area EDC's!) required, digital EDC's and control systems may well take over a large share of the market. Digital computers permit accurate total-cost comparisons of different system configurations, including start-up and shut-down costs, and are almost competitive in price with analog EDC's.

References

1. Steinberg, M. I., and T. H. Smith: *Economic Loading of Steam Power Plants and Electric Systems*, John Wiley & Sons, Inc., New York, 1943.
2. George, E. E.: Intrasystem Transmission Losses, *Trans. AIEE*, **62**:153–158 (March, 1943).
3. Ward, J. B., J. R. Eaton, and H. W. Hale: Total and Incremental Losses in Power Transmission Network, *Trans. AIEE*, **69**(1):626–631 (1950).
4. Kron, G.: Tensorial Analysis of Integrated Transmission Systems, *Trans. AIEE*, **70**(1):1239–1248 (1951).

5. Kirchmayer, L. K., and G. W. Stagg: Analysis of Total and Incremental Losses in Transmission Systems, *Trans. AIEE*, **70**(1):1197–1205 (1951).
6. Glimn, A. F., R. Habermann, Jr., L. K. Kirchmayer, and G. W. Stagg: Loss Formulas Made Easy, *Trans. AIEE*, **72**(III):730–735 (1953).
7. Early, E. D., R. E. Watson, and G. L. Smith: A General Transmission Loss Equation, *Trans. AIEE*, **74**(III): 510–516 (1955).
8. Morrill, C. D., and J. A. Blake: A Computer for Economic Scheduling and Control of Power Systems, *Trans. AIEE*, **74**(III):1136–1141 (1955).
9. Squires, R. B., H. W. Calborn, R. T. Byerly, and W. R. Hamilton: Loss Evaluation, Part V: Economic Dispatch Computer Design, *Trans. AIEE*, **75**(III):719–727 (1956).
10. Ehrenberg, Gustave: Automatic Load-frequency Control System for Central Station Power, *Trans. AIEE*, **74**(III):787–795 (1955).
11. Burnett, K. N., D. W. Halfhill, and B. R. Shepard: A New Automatic Dispatching System for Electric Power Systems, *Trans. AIEE*, **75**(III):1049–1054 (1956).
12. Kompass, E. J.: The "Early Bird" Goes Automatic, *Control Eng.*, December, 1956, pp. 77–83.
13. Kirchmayer, L. K.: *Economic Operation of Power Systems*, John Wiley & Sons, Inc., New York, 1958.

Section 6

ELECTRONIC ANALOG COMPUTERS: SPECIAL COMPONENTS AND TECHNIQUES

By

JEROME D. KENNEDY, Sr., *Vice-President, Applied Dynamics, Inc., Ann Arbor, Mich.*

PAUL E. RUSSELL, *Professor and Head, Department of Electrical Engineering, University of Arizona, Tucson, Ariz.*

GRANINO A. KORN, *Professor of Electrical Engineering, University of Arizona, Tucson, Ariz.*

W. K. McGREGOR, *ARO, Inc., Tullahoma, Tenn.*

R. M. LEGER, *Chief, Flight Performance and Analysis, General Dynamics/Astronautics, San Diego, Calif.*

JEROME L. GREENSTEIN, *Senior Research Engineer, General Dynamics/Astronautics, San Diego, Calif.*

L. D. KOVACH, *Design Specialist, Douglas Aircraft Company, Inc., El Segundo, Calif., and Professor of Mathematics, Pepperdine College.*

CONTENTS

PART 1: REPRESENTATION OF TIME DELAYS

Jerome D. Kennedy, Sr.

Introduction

6.1.1. Time Delays 6–3
6.1.2. Requirements 6–3

Lumped-parameter Approximations

6.1.3. Lumped-parameter Approximations Using Operational Amplifiers 6–5
6.1.4. Higher-order Approximations 6–5

Capacitor-storage Methods

6.1.5. Capacitor Sample-hold Storage 6–8
6.1.6. Relay-controlled Capacitor Storage 6–8
6.1.7. Capacitor Wheel 6–8

Tape-recording Systems

6.1.8. Analog Tape 6–10
6.1.9. Amplitude Modulation (Direct Recording) 6–10
6.1.10. Frequency Modulation 6–11
6.1.11. Pulse-width Modulation 6–12
6.1.12. Digital Tape 6–13

Other Recording Schemes

6.1.13. Perforated Tape 6–14
6.1.14. Magnetic-drum Storage 6–14
6.1.15. Analog Drum 6–14
6.1.16. Digital Drum 6–14
6.1.17. Use of Servo Tables 6–16

PART 2: REPETITIVE ANALOG COMPUTERS

Paul E. Russell

Introduction

6.2.1. Repetitive Analog Computers. 6–17
6.2.2. Repetitive vs. "Slow" Electronic Analog Computers..... 6–17
6.2.3. Applications..... 6–17

Special Components

6.2.4. Survey..... 6–20
6.2.5. Amplifiers..... 6–20
6.2.6. Reset Circuits..... 6–22
6.2.7. Control Unit and Displays.... 6–24

PART 3: ANALOG STORAGE TECHNIQUES FOR ELECTRONIC ANALOG COMPUTERS

Granino A. Korn

Analog Storage Circuits and Applications

6.3.1. Introduction..... 6–26
6.3.2. Sample-hold Circuits..... 6–26
6.3.3. Accurate Electronic Switches. 6–26
6.3.4. Time Sharing (Multiplexing) of Computing Elements..... 6–27
6.3.5. Finite-difference Techniques. 6–29

Special Computers Employing Analog Sampling Techniques

6.3.6. Stepping-relay Techniques... 6–29
6.3.7. A Hybrid Analog-digital Computer..... 6–29

Sampled-data Techniques for Repetitive-analog-computer Read-out and Statistical Computations

6.3.8. Use of Sample-hold Circuits for Repetitive-analog-computer Read-out..... 6–31
6.3.9. Repetitive-analog-computer Estimation of Ensemble Statistics in Random-process Studies..... 6–31
6.3.10. Computation of Time Averages for Stationary Random Processes..... 6–31
6.3.11. An Advantage of Sampled-data Techniques for Statistical Computations..... 6–32

Automatic Sequential Computation

6.3.12. Systems and Components.... 6–32
6.3.13. Applications..... 6–33

PART 4: ELECTRONIC ANALOG COMPUTERS AS CONTROL-SYSTEM COMPONENTS

W. K. McGregor

Introduction

6.4.1. Survey..... 6–35

Analog-computing Elements in Direct Control

6.4.2. Linear Operations..... 6–35
6.4.3. Nonlinear Operations..... 6–36
6.4.4. A Specific Application: Gas-flow Process..... 6–36
6.4.5. Computer Requirements..... 6–39

Computers in Indirect Control

6.4.6. Indirect Control..... 6–39

Concluding Remarks

6.4.7. Economic Considerations..... 6–40
6.4.8. By-products of Analog Control 6–40
6.4.9. Suggested Developments..... 6–41

PART 5: COMBINED USE OF ANALOG AND DIGITAL COMPUTERS

R. M. Leger and Jerome L. Greenstein

Introduction

6.5.1. Analog and Digital Computation..... 6–42
6.5.2. Combined Analog-digital Simulation..... 6–42

Numerical Techniques Used in Digital Simulation

6.5.3. Sampled-data Interpolation.. 6–43
6.5.4. Digital Solution of Differential Equations..... 6–43

Conversion Equipment

6.5.5. Digital-to-analog Converters (Decoders)..... 6–44
6.5.6. Analog-to-digital Converters (Encoders)..... 6–44
6.5.7. Analog Sample-hold Circuits. 6–44
6.5.8. Buffering of Digital Data.... 6–44
6.5.9. Synchronization..... 6–45
6.5.10. Conversion Errors..... 6–46

Problems Peculiar to Combined Analog-digital Simulation

6.5.11. Sampling Problems..... 6–46
6.5.12. Desampling..... 6–46
6.5.13. Digital-computer Speed and Time Delay..... 6–48
6.5.14. Simulation Setup..... 6–49

Example of a Combined Analog-digital Simulation

6.5.15. A Missile-system Problem... 6–49
6.5.16. Combined Simulation..... 6–50
6.5.17. Sequence of Computer Operations..... 6–51
6.5.18. The Digital Computer as a Function Generator..... 6–52

PART 6: MISCELLANEOUS TECHNIQUES

L. D. Kovach

Solution of Difference Equations

6.6.1. Difference Equations..... 6–52

Computation of Fourier Integrals without Analog Multipliers

6.6.2. A Weighting-function Technique..... 6–54
6.6.3. Inverse Fourier Transforms... 6–55
6.6.4. Harmonic Analysis of Periodic Functions..... 6–56

Conformal Mapping

6.6.5. Theory..... 6–56
6.6.6. An Airfoil Transformation.... 6–56

PART 1: REPRESENTATION OF TIME DELAYS

By Jerome D. Kennedy, Sr.

INTRODUCTION

6.1.1. Time Delays. Many analog-computer applications require representation of devices which introduce a *time delay*. If the input to such a device is a function $f(t)$ of the time t, the output is the delayed function $f(t - \tau)$. Suitable implementation of such delays can appreciably extend the utility of an electronic analog computer, e.g., in the following representative applications (see also Secs. 5.7, 5.8, and 5.9):

Transmission of radar or sonar signals, ocean waves, and nuclear radiation

Pipeline transport delays in process-control applications,[1] jet-engine and rocket systems, rocket-combustion lags,[2] nuclear power plants,[3] human circulatory systems

Movement of materials through paper mills,[4] steel mills,[5] and moving-vehicle wheel displacements

Correlation-function computations (Sec. 5.9)

6.1.2. Requirements. Time-delay-unit specifications, apart from cost, include the following:

Range of delay time (less than a second, seconds, minutes, hours)

Variability of delay (fixed, stepped, continuous)

Number of channels (same function with different delay times, several functions of same delay time)

Storage capacity (maximum time delay—frequency product $\tau\omega$, number of samples per cycle of signal)

Accuracy (amplitude attenuation, phase-shaft error, delay accuracy)

Compatibility with computer (voltage level, input-output impedance, signal form, AM, FM, PWM, PCM, etc.)

The ideal time-delay device will exhibit *zero attenuation and a phase lag accurately proportional to frequency.* Cost and complexity, plus the characteristics of the allied computing equipment, will indicate allowable compromises with the ideal. Some physical processes require a controlled *variable* delay. Correlation problems will permit a "stepped" delay.

Table 6.1.3 lists various time-delay devices and their characteristics.

Table 6.1.1. Comparison of Time-delay Devices

Storage element	Analog tape			Digital tape		Magnetic drum		Capacitor storage			Active element	Mechanical plotter	
Storage technique	FM		PWM	Mag-netic	Perfo-rated	Analog	Digital	Memory wheel	Bucket brigade	AM	Operational-amplifier approx.	Curve follower	
Characteristic \ Art.	6.1.10		6.1.11	6.1.14	6.1.13	6.1.15	6.1.16	6.1.7	6.1.5	.1.9	6.1.4	6.1.17	
Data:													
Continuous	X	X	X	X	X	X	X	—	—	X	X	X	X
Sampled	X	X	X	X	X	X	X	X	X	X	—	X	X
Analog	—	—	—	X	X	X	X	X	X	X	X	—	—
Digital	1	1	0.5	X 0.01	X 0.5	X 0.03	X 0.05	1	1	1	0.1	0.2	—
Static accuracy, % full scale	X (X)	X (X)	X —	— —	— —	— —	— —	X X X	X X X	X (X) —	0.25	—	—
Delay:													
Fixed	No data	No data	X 1	X No data	X 1.67	X No data	X 0.5	1	No data	No data	0–11	X 0.1	X No data
Stepped	—	—	0.04–1.5	No data	3–1800	5–20	0.1–200	1/30–10	No data	—	—	—	—
Continuously variable	X (loop)	X (loop)	X (loop)	X (loop)	X	X	X	X	X	X (loop)	X	X	X
Accuracy, % full scale	X (reel)	X (reel)	X (reel)	X (reel)	X (reel)	—	X	—	—	X (reel)	—	X	X
Range, sec	*	*	*	*	*	No data	644†	188†	No data	*	7.5	X	No data

* $\tau\omega$ will be determined by head spacing and data density.
† $\tau\omega$ is based on 10 samples per cycle, which results in an rms error of 2 to 5 per cent.
X Applicable.
(X) Alternate.
— Not applicable.

LUMPED-PARAMETER APPROXIMATIONS

6.1.3. Lumped-parameter Approximations Using Operational Amplifiers. In Laplace-transform notation pure time delay is expressed by the transfer function

$$G(s) = e^{-\tau s} \tag{6.1.1}$$

so that the frequency-response function becomes

$$\begin{aligned} G(j\omega) &= \cos \omega\tau - j \sin \omega\tau \\ &= 1/\underline{-\omega\tau \text{ radians}} \end{aligned} \tag{6.1.2}$$

Equation (2) calls for a circuit with zero attenuation and phase shift proportional to frequency. No lumped-parameter networks exhibit these characteristics exactly; the simplest approximation is

$$G(s) = \frac{1 - \tau s/2}{1 + \tau s/2} \approx e^{-\tau s} \tag{6.1.3}$$

which is implemented by the computer setup of Fig. 6.1.1*a*.[8] The phase shift of this circuit departs noticeably from a linear relationship with frequency for $\tau\omega = 0.6$ radian. Thus, for a signal frequency ω of 6.28 radians per sec the time delay τ must be less than 0.1 sec to prevent a phase error greater than 1 deg.

6.1.4. Higher-order Approximations. Higher-order zero-attenuation approximations to the function $e^{-\tau s}$ can be written in the form

$$G(s) = \prod_{i=1}^{n} \frac{\tau_i s^2 - 2\zeta_i \tau_i s + 1}{\tau_i s^2 + 2\zeta_i \tau_i s + 1} \approx e^{-4\sum_{i=1}^{n} \zeta_i \tau_i s} = e^{-\tau s} \tag{6.1.4}$$

where the choice of the coefficients (damping ratios) ζ_i will depend on the error criterion (phase error, output noise level, step-function response, etc.) employed. For sinusoidal inputs the phase shift is given by

$$\phi_{\text{actual}} = 2 \sum_{i=1}^{n} \arctan \frac{2\zeta_i \tau_i \omega}{1 - \tau_i^2 \omega^2} \tag{6.1.5}$$

so that the phase error associated with the approximation (4) is

$$\delta = \phi_{\text{actual}} - \phi_{\text{ideal}} = 2 \sum_{i=1}^{n} \arctan \frac{2\zeta_i \tau_i \omega}{1 - \tau_i^2 \omega^2} - \tau\omega \tag{6.1.6}$$

The so-called *Padé approximation*[10,11] frequently used to determine the damping ratios ζ_i in Eq. (4) yields a close approximation for very low frequencies but is poor at higher frequencies and hence yields a poor step-function response. The *cut-product approximation* of Warfield and Weaver[12,13] yields a better step-function response. A practical approach is that of Single,[9] who plotted phase-error terms $\arctan 2\zeta_i \tau_i \omega/(1 - \tau_i^2\omega^2)$ for various values of ζ_i (Fig. 6.2.2) and picked $\zeta_1, \zeta_2, \ldots, \zeta_n$ by trial and error so as to minimize the total phase error (6).

For $\omega\tau < 1.5$ radians, the computer setup of Fig. 6.1.1*b*[9] implements Eq. (4) for $n = 1$, with

$$\tau_1 = R\sqrt{C_{a1}C_{b1}} \qquad \zeta_1 = \sqrt{\frac{C_{b1}}{C_{a1}}} \qquad R = \frac{\psi_1 R_0}{4\zeta_1^2} \qquad \tau = 4\zeta_1\tau_1 \tag{6.1.7}$$

$\zeta_1 = \sqrt{2}/2$ is a favorable choice. The computer setup of Fig. 6.1.1*c* is more suitable for use with general-purpose analog computers.

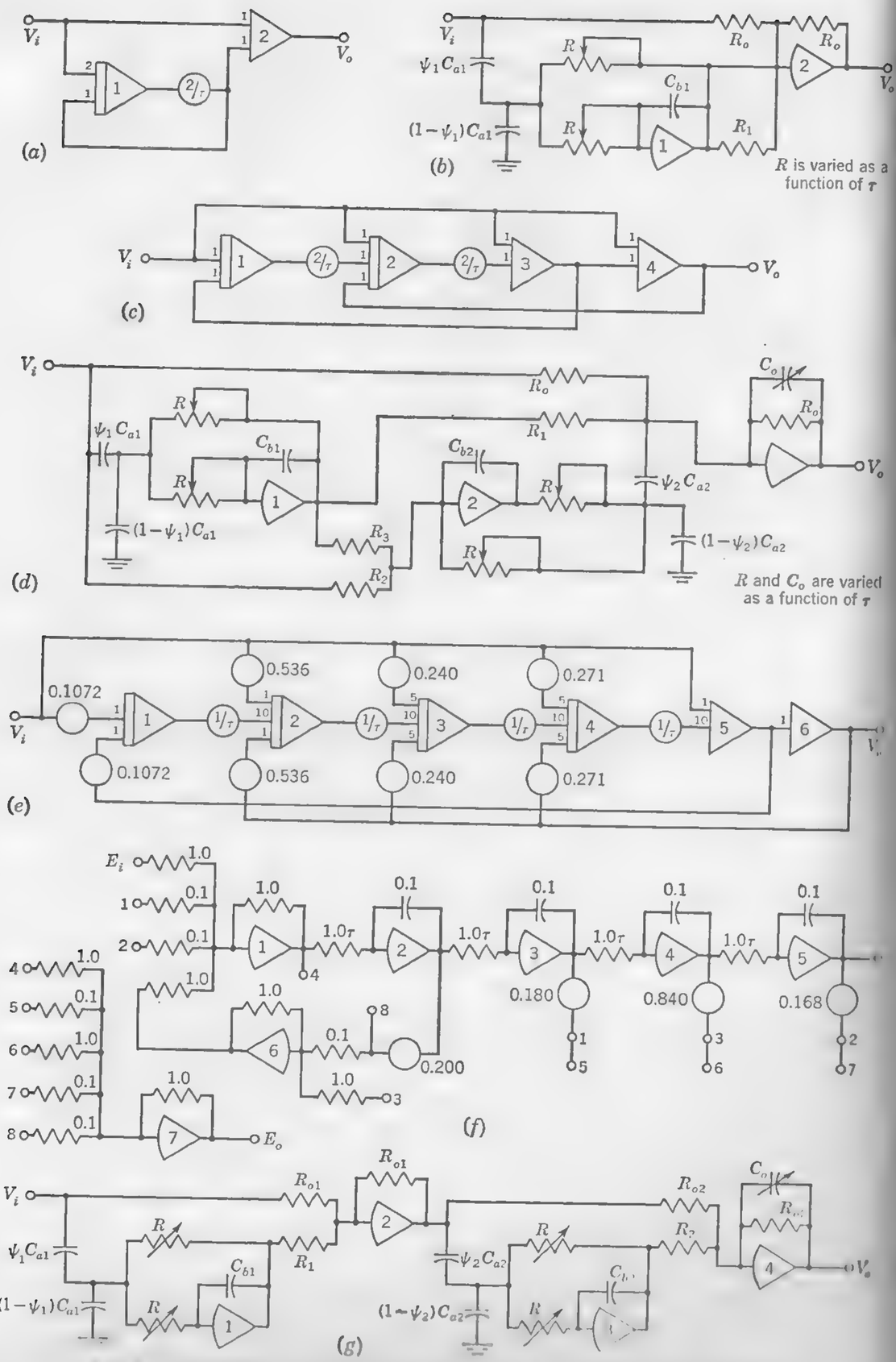


FIG. 6.1.1. Lumped-parameter approximations for a fixed time delay (see text).

To improve noise level and step-function response it is best to abandon zero-attenuation approximations.[11,14] Figure 6.1.1*d*[9] shows a circuit with the transfer function

$$G(s) = \frac{1}{\tau_0 s + 1} \cdot \frac{\tau_1 s^2 - 2\zeta_1\tau_1 s + 1}{\tau_1 s^2 + 2\zeta_1\tau_1 s + 1} \cdot \frac{\tau_2 s^2 - 2\zeta_2\tau_2 s + 1}{\tau_2 s^2 + 2\zeta_2\tau_2 s + 1} \approx e^{-\tau s} \qquad (6.1.8)$$

with delay time $\tau = 4(\zeta_1\tau_1 + \zeta_2\tau_2) + \tau_0$ and

$$\begin{aligned} \tau_1 &= R\sqrt{C_{a1}C_{b1}} \qquad \zeta_1 = \sqrt{\frac{C_{b1}}{C_{a1}}} \qquad R = \frac{\psi_1 R_0}{4\zeta_1^2} \\ \tau_2 &= R\sqrt{C_{a2}C_{b2}} \qquad \zeta_2 = \sqrt{\frac{C_{b2}}{C_{a2}}} \qquad R = \frac{\psi_2 R_0}{4\zeta_2^2} \\ R_3 &= \frac{\psi_1\psi_2 R_0}{16(\zeta_1\zeta_2)^2} \qquad \tau_0 = R_0 C_0 \\ \zeta_1 &= \sqrt{3/2} \qquad \zeta_2 = 0.4 \qquad \frac{\tau_1}{\tau_2} = 1.68 \qquad \tau_0 \le \frac{\tau}{100} \end{aligned} \qquad (6.1.9)$$

Figures 6.1.1*e* and 6.1.1*f* show analogous circuits suitable for general-purpose analog computers. The circuit of Fig. 6.1.1*f* saves a number of potentiometers at the expense of an additional amplifier.

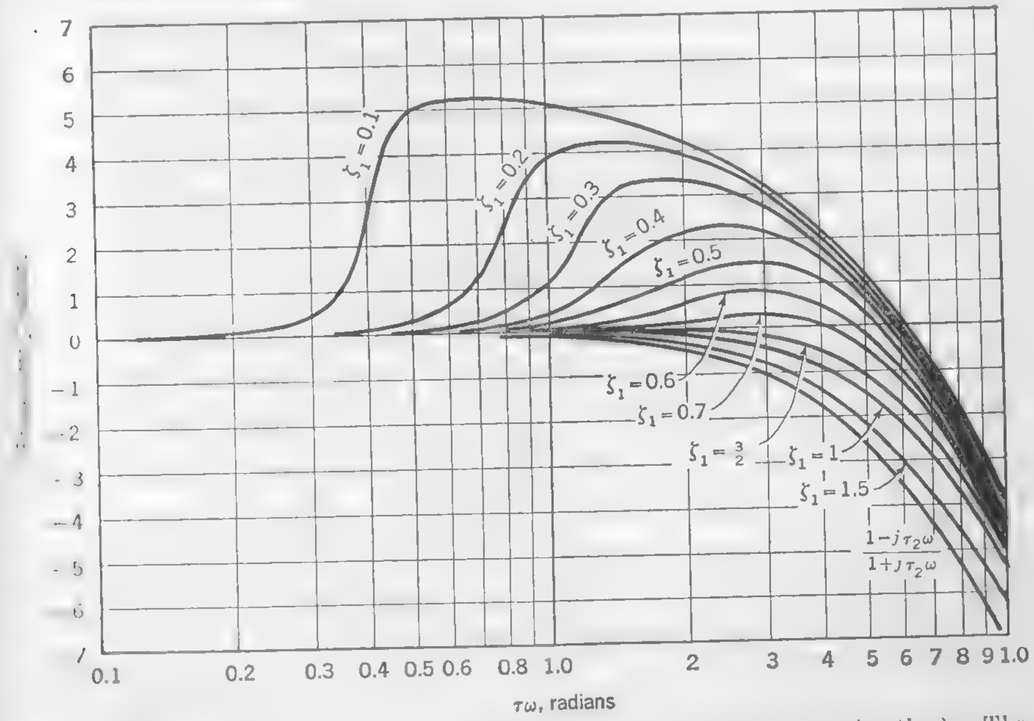


FIG. 6.1.2. Phase error vs. $\tau\omega$ for Eq. (4) with $n = 1$ (second-order approximation). The phase error for a first-order approximation is also shown for comparison (from Ref. 9).

The circuit of Fig. 6.1.1*d* has a somewhat high output noise level for delays τ below 1 sec. The rms noise level is approximately $0.040/(\tau + 0.040)$ volts, mostly at 60 cps. For further noise-level reduction an additional amplifier and passive elements may be added for filtering purposes (Fig. 6.1.1*g*).[9] The noise level for small time delays is about 20 mv rms.

For longer delays one can cascade, say, three circuits like that shown in Fig. 6.1.1*d* to obtain time delays between 0 and 33 sec; Fig. 6.1.3 illustrates the step-function response of such a circuit.[9] Time-delay units of this type are commercially available.[15]

CAPACITOR-STORAGE METHODS

6.1.5. Capacitor Sample-hold Storage. Time delays can be realized through storage of successive samples of the input function on individual capacitors whose voltages are read after a suitable delay. This technique has been implemented with the aid of a commutator wheel,[15,16] stepping switches (see also Sec. 6.3),[17] relays,[18] and electronic switches.[19] The output of each device is a time series of impulse voltages, and the delay is the product of the average dwell time per stage (capacitor) and the number of stages between input and output. To eliminate distortion, the sampling frequency must be determined, and if the input signal contains frequency components higher than one-half the sampling rate,[20] filtering is required. Since the sampled data

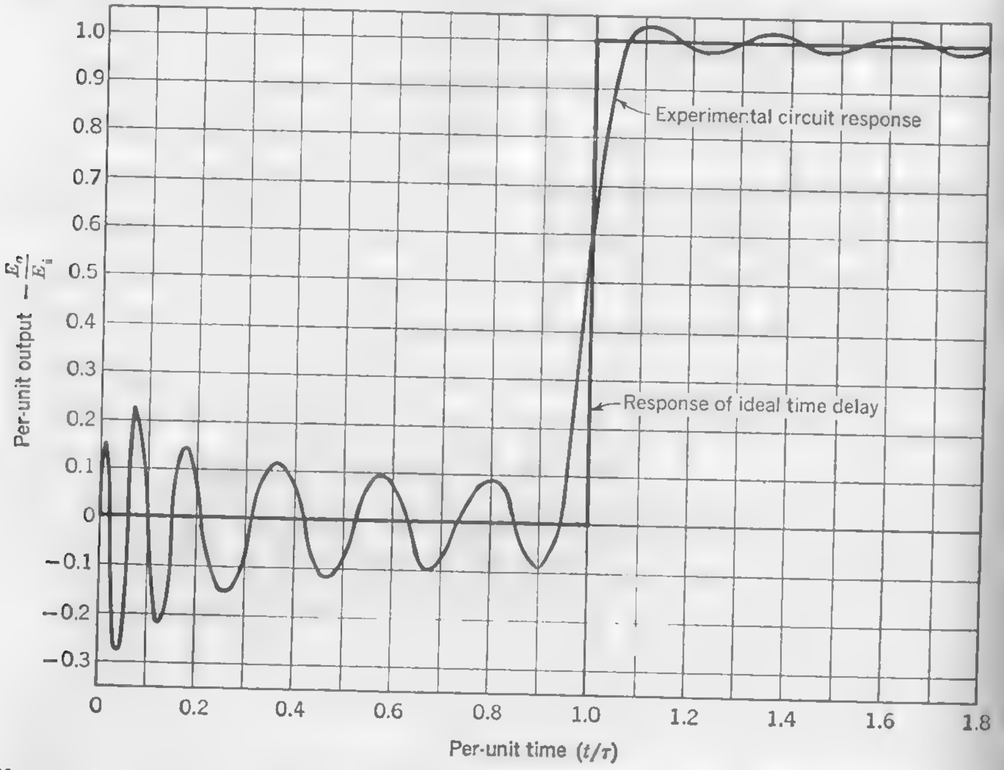
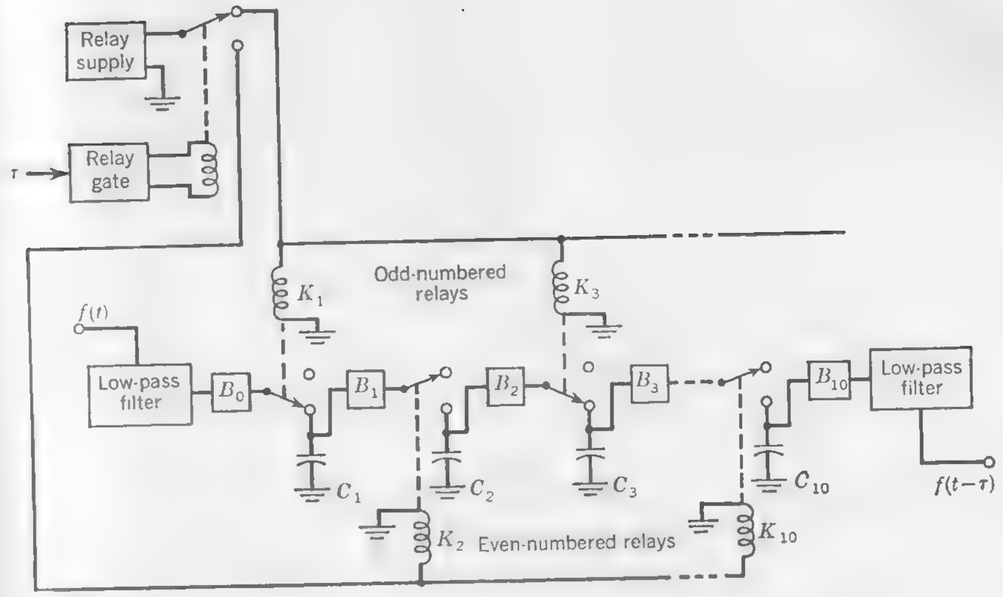


FIG. 6.1.3. Step-function response for three circuits of Fig. 6.1.1*d* in cascade (from Ref. 9).

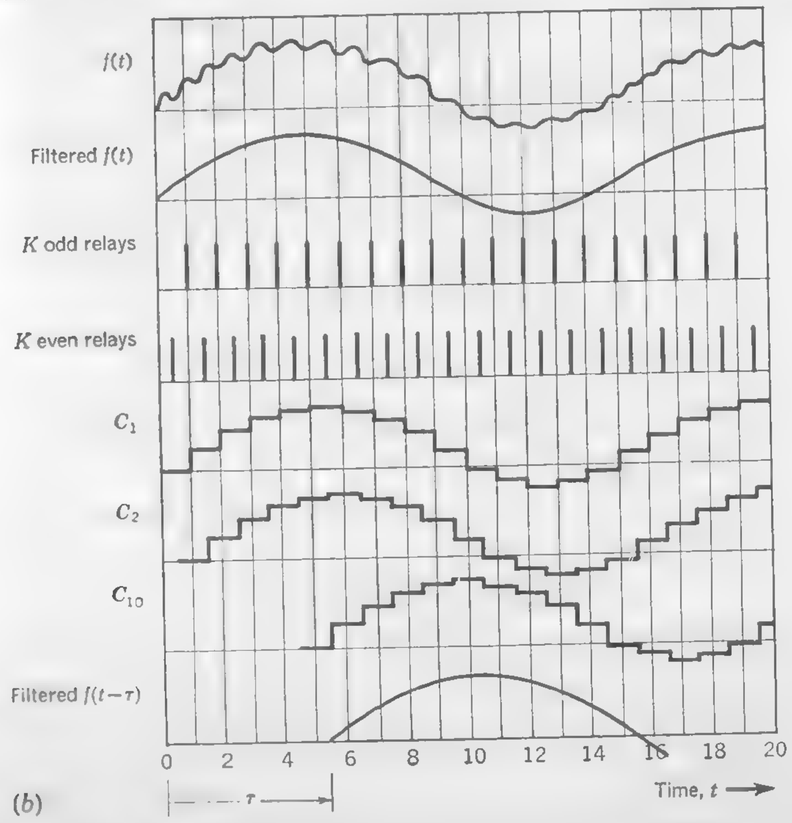
are a series of impulses at the output stage, filtering is again required to reduce distortion in the load circuit. The most common type of filter involves a simple holding circuit which emits a staircase voltage (see also Sec. 6.3).

6.1.6. Relay-controlled Capacitor Storage. Switching voltages from one capacitor to the next may be accomplished with relays and buffer amplifiers which charge the succeeding capacitors in one mode and reproduce the previous capacitor voltage without draining off charge in the other mode. Thus, the four requirements for an ideal buffer are unity gain, infinite input impedance, zero output impedance, and infinite output power. Figure 6.1.4 illustrates the circuit and waveforms; note that the odd relays are on while the even relays are off, and vice versa. Even and odd relays must never contact the capacitor terminals simultaneously.

6.1.7. Capacitor Wheel. Canadian Westinghouse Co., Ltd.,[16] has developed a capacitor-storage time delay around a rotating commutator 14 in. in diameter carrying 300 Mylar 0.1-μf capacitors (Figs. 6.1.5 and 6.1.6). The commutator bars terminate one end of the capacitors, while the hub provides a common ground for the other ends. The wheel runs at a constant speed, and the delay is varied by servo positioning of the



(*a*)



(*b*)

FIG. 6.1.4. Relay-operated capacitor-storage delay (*a*) and waveforms (*b*).

read-out brush; the time delay is proportional to the spacing between read and write brushes.

The output clamp consists of a Mylar 0.01-μf capacitor. For sharper rise times the capacitor should be changed to 0.001 μf. In order to present a high impedance to the capacitors, a pair of operational amplifiers are used as a stabilized follower circuit (Fig. 6.1.6*c*); an integrator sample-hold circuit (Sec. 6.3) could be used instead.[15]

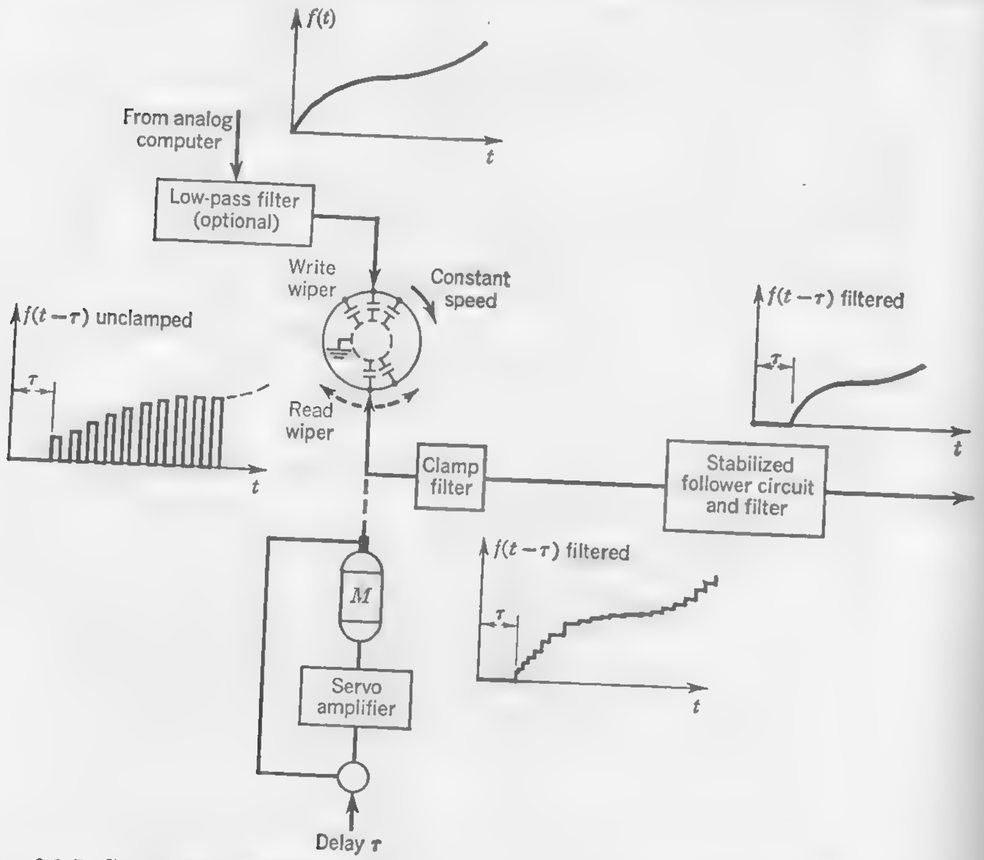


FIG. 6.1.5. Canadian Westinghouse capacitor-wheel memory. *M* is a combination motor-tachometer used in a rate-servo circuit.

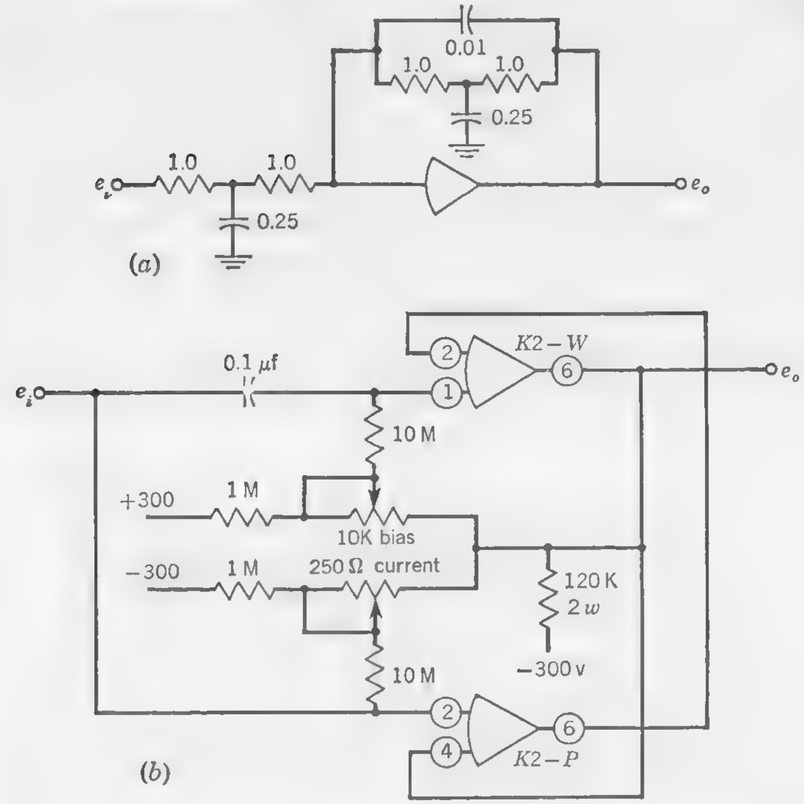
Specifications for the memory are included in Table 6.1.1, and the frequency response is shown in Fig. 6.1.6*d*.

TAPE-RECORDING SYSTEMS

6.1.8. Analog Tape. Magnetic flux storage on metallic or oxide tape is another method of creating an analog time delay. One converts an electrical signal to magnetic flux, charges the moving magnetic tape with this flux, and reads the signal voltage as the charged tape passes by a reading head. The distance between the recording and reading head divided by the average tape speed yields the signal delay τ. One advantage of magnetic tape, regardless of the recording method, is its long storage capacity. In this connection, a *tape loop* permits reuse of a small amount of tape, but old data are lost after every tape cycle.

6.1.9. Amplitude Modulation (Direct Recording). Direct recording is limited in accuracy (1 per cent) and frequency response (10 cps lower cutoff). Furthermore, any change in the tape speed increases the reproduced signal level, so that speed-sensitive gain-control circuits must be employed to hold the 1 per cent figure. Thus direct-recording magnetic-tape recorders are of doubtful value for electronic-analog-computer time-delay generation.

6.1.10. Frequency Modulation. With a frequency-modulated-carrier technique for entering electrical data on magnetic tape accuracy to 0.1 per cent is achievable down to zero frequency. One reduces *flutter* due to rapid tape-speed changes by adding high inertia to the tape-drive mechanism. *Wow*, caused by slowly changing tape



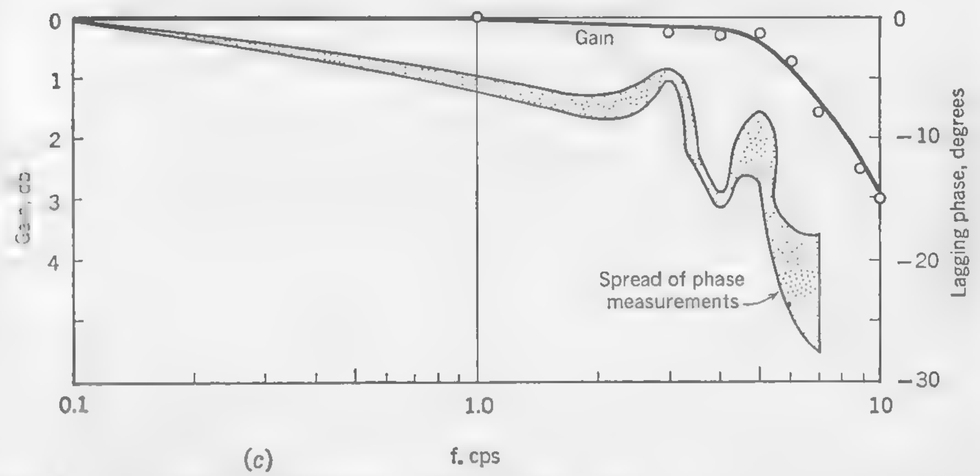


FIG. 6.1.6. Canadian Westinghouse capacitor-wheel memory: input filter (*a*), stabilized follower amplifier (*b*), and frequency response (*c*). The follower circuit of Fig. 6.1.6*c* was originally designed by D. Dedford of Northwestern University.

speed, can be compensated for by a fixed-frequency reference signal recorded on a separate tape track. Changes in this comparison frequency due to flutter are then used to change the center (demodulation) frequency of the data-channel discriminator. A *variable* time delay is almost impossible to obtain because changes in tape speed or head position would affect the *amplitude* of the reproduced signal.

6.1.11. Pulse-width Modulation.[21] Electronic multipliers of the time-division type employ pulse-width–pulse-amplitude modulation (Sec. 3.2); Electronic Associates, Inc., modified such an electronic multiplier for the Simulag Group Model 1.002. The research laboratories of the General Motors Corporation[22] employ this time delay as a "road-function generator" to provide forcing functions in four-wheel-vehicle simulation. The road-displacement signal (previously stored on the tape for left and right wheels) is played back into a pair of tandem-read heads. This distance between the tandem heads represents wheel base, and the tape velocity is analogous to vehicle speed. Tape speed is controlled by a voltage generated in the suspension simulator (general-purpose electronic analog computer), or by a potentiometer on the control panel.

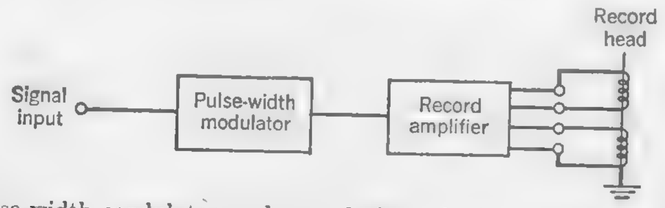


FIG. 6.1.7. Pulse-width modulator and record channel of Simulag Group Model 1.002. (*Electronic Associates, Inc., Long Branch, N.J.*)

The system can also be used to provide a time delay without recourse to a previously prepared tape. To do this, a signal is recorded using the lower head and picked off by the upper head. In this process, signal-frequency translations occur only during periods when the tape is being accelerated. The reels hold 3,600 ft of magnetic tape, giving a minimum playing time (at 40 in. per sec) of 18 min. System performance characteristics are given in Table 6.1.1. The maximum frequency which may be recorded is 30 cps, but during playback at higher tape speed frequencies up to 100 cps can be accurately reproduced.

Figure 6.1.7 shows one record channel. The signal is converted at a pulse-repetition rate of 500 cps into a constant-amplitude width-modulated pulse having a duty cycle of from 20 per cent for −100-volt input to 80 per cent for +100 volts. This process adds a d-c component to the signal, which is later removed. The pulse information is applied in push-pull fashion to a record amplifier comprising a constant-current source and an electronic switch. This controls the currents in the balanced record-head windings.

The reproduce process is shown in Fig. 6.1.8. Push-pull signals, generated in differentiated form by the reproduce head at a level of several millivolts, are amplified in separate class A channels up to a few volts. They are then half-wave-rectified to 100-volt pulses and applied to a clipping circuit which produces output pulses when the 100-volt pulses reach about 50 volts. These pulses are further amplified, still in separate channels, and finally operate on the bistable multivibrator to reproduce the pulse-width modulated input wave. Important portions of this process are shown in the center portion of Fig. 6.1.8, where the short horizontal lines represent the clipping level referred to the input signals. These signals, incidentally, increase in magnitude by a factor of 10 when the tape speed is increased from 4 to 40 in. per sec, yet the system maintains its accuracy without use of volume control or age circuitry.

The pulse-width-modulated wave from the multivibrator is squared and amplitude-limited; then the false d-c component is subtracted out. The signal is then passed through a rather complicated low-pass filter associated with the output amplifier.

The tape-speed-control system is a rate servo controlling the capstan-drive motor (Fig. 6.1.9). The Ampex FR 1100 tape transport is modified by substituting a spe-

cial induction-motor–d-c-tachometer combination for the standard hysteresis synchronous motor, and coupling a d-c tachometer generator directly to the motor. The response time of the speed-control system to a control-voltage step function of 0 to 100 volts or 100 volts to 0 is less than 3 sec; 100 volts corresponds to 40 in. per sec.

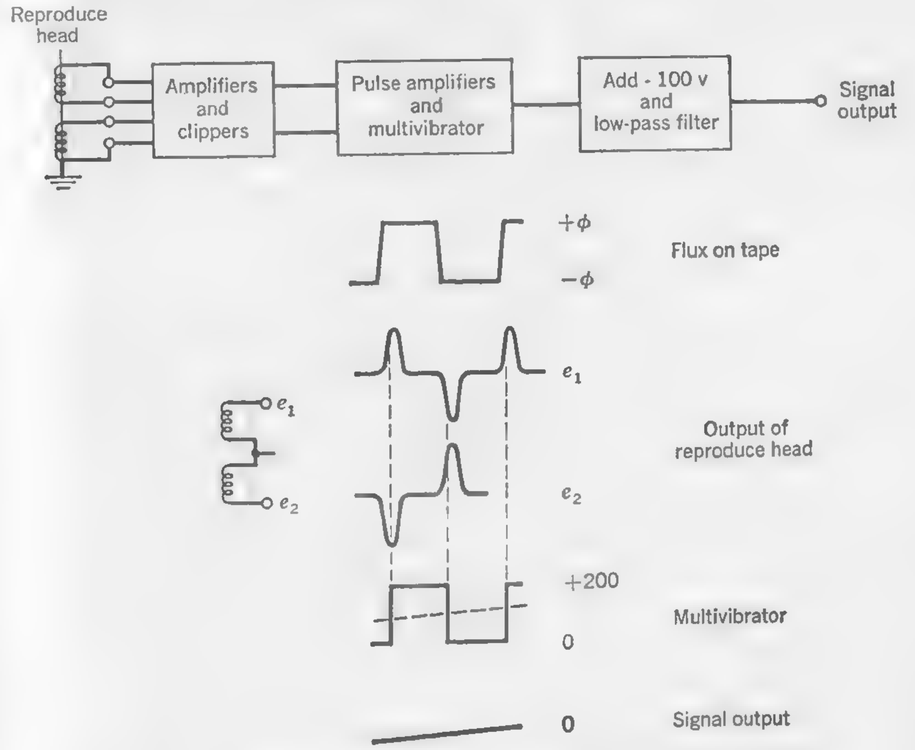


FIG. 6.1.8. Reproduce process in Simulag Group Model 1.002.

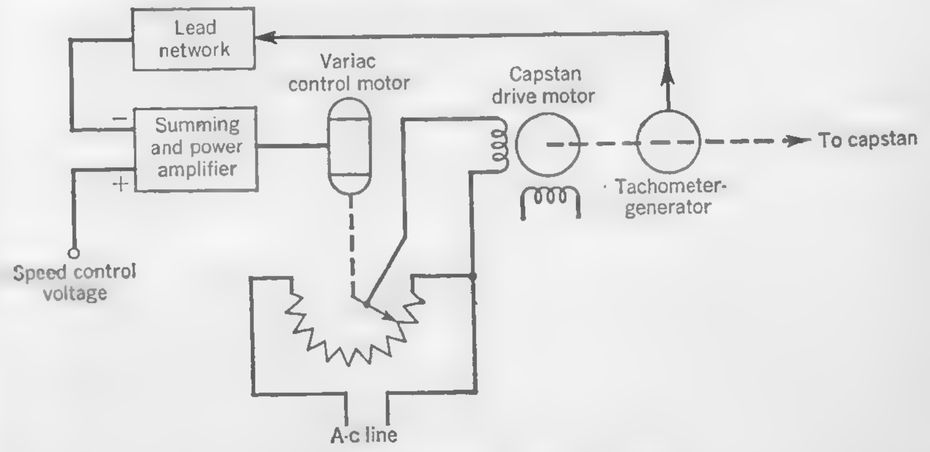


FIG. 6.1.9. Tape-speed control system.

6.1.12. Digital Tape. Pulse-code modulation, which is used to prepare punched tape, is a fourth possible method of storing data on magnetic tape. In essence, an analog-to-digital converter transforms the instantaneous value of an analog voltage into a digital code such as binary or binary coded decimal. The code groups are later read off the tape and restored to analog form by digital-to-analog converters with appropriate filtering. Here again head spacing and tape speed determine the delay.

A very elaborate system of this type is described in Ref. 24. It may also be worthwhile to glance at the possibilities of adapting a well-known digital magnetic-tape unit. The IBM 727 is an input-output tape unit for the IBM 704 electronic data processing machine. Binary code is quite common, and a 36-bit word is stored on the magnetic tape in a 6 × 6 format, i.e., six tracks (plus a parity check track) across the tape and six characters along the tape. The maximum tape speed is 75 in. per sec and the data density is 200 characters per inch. Thus, the word rate is $75 \times 200 = 2{,}500$ words per second. If half-words (18 bits) were used, the word rate would be 5,000. Space must be allowed for word and end-of-record gaps so the actual word rate might be reduced to 4,000 words per second. If 14 binary bits out of the 18 were used for amplitude, one could use a commercially available analog-to-digital converter for loading the tape and an all-electronic digital-to-analog converter (accurate to 14 binary bits) to restore the data to analog form.

OTHER RECORDING SCHEMES

6.1.13. Perforated Tape. A punched-tape delay simulator has been used by D. A. Phlegel and D. M. Vesper (Phillips Petroleum Company, Bartlesville, Okla.) with general-purpose electronic analog computers to simulate the lags encountered in refinery-plant control. Eight binary bits are punched across the tape, which provides a resolution of 1 part out of 511. By punching a complete word across the tape a maximum word rate within the limits of the tape punch and reader is realized. Specifications for the punched-tape delay simulator are shown in Table 6.1.1. The analog-to-digital and digital-to-analog converters were designed at Phillips Petroleum Company.

6.1.14. Magnetic-drum Storage. The magnetic drum is another common digital storage device which has been exploited in the search for suitable time-delay devices. Like magnetic tape, but limited in storage capacity by drum circumference and data density, the signal may be stored on the drum in analog or digital form by charging the metallic surface with magnetic flux. Restoring the recorded flux at a later time to original signal amplitude constitutes the delay. The delay can be varied by changing drum speed of a slowly rotating drum (analog) or varying the read-command period for high-speed drums (digital). Table 6.1.1 shows comparative characteristics of two magnetic-drum delay systems described below.

6.1.15. Analog Drum. Sampson and Daniel[23] used a low-speed drum employing rotating scanners for record and playback. In the record mode a spot of flux is applied to the surface of the drum at a distance from the edge proportional to the amplitude of the function to be delayed. Similarly, during the playback, the flux position is compared with a voltage ramp (sawtooth) which is initiated as the playback scanner head crosses the edge of the drum. Thus the instantaneous amplitude of the delayed signal is equal to the sawtooth voltage at the instant the playback head crosses the flux spot. The playback sawtooth is initiated with the aid of a prerecorded reference track along the leading edge of the drum; the reference track generates trigger pulses to reset the sawtooth.

6.1.16. Digital Drum. Electronic Associates, Inc., Magnetic Drum Simulag Group Model 1.011 stores quantized data in binary form on a magnetic drum with subsequent reconstruction to analog form; the dwell time on the drum is the time delay τ. This delay τ is controlled by an electronic ratchet technique which precesses the address of the quantized data one position on the drum for every sample. After a voltage-controlled number of samples the digital data are sequentially read off the drum at the same sampling rate. Figure 6.1.10 shows an over-all diagram of the system.

The quantizing device is an all-solid-state analog-to-digital converter which, upon command, converts an analog voltage to a 12-bit binary word and stores it in the buffer register. The conversion period is less than 1 msec, and the binary data are available in parallel form for storage on 12 tracks of the drum.

The magnetic drum is 5 in. in diameter, rotates at a speed of 120 rps, and has a capacity of 1,024 bits per track.

The digital-to-analog converter is connected at all times to the read amplifiers and

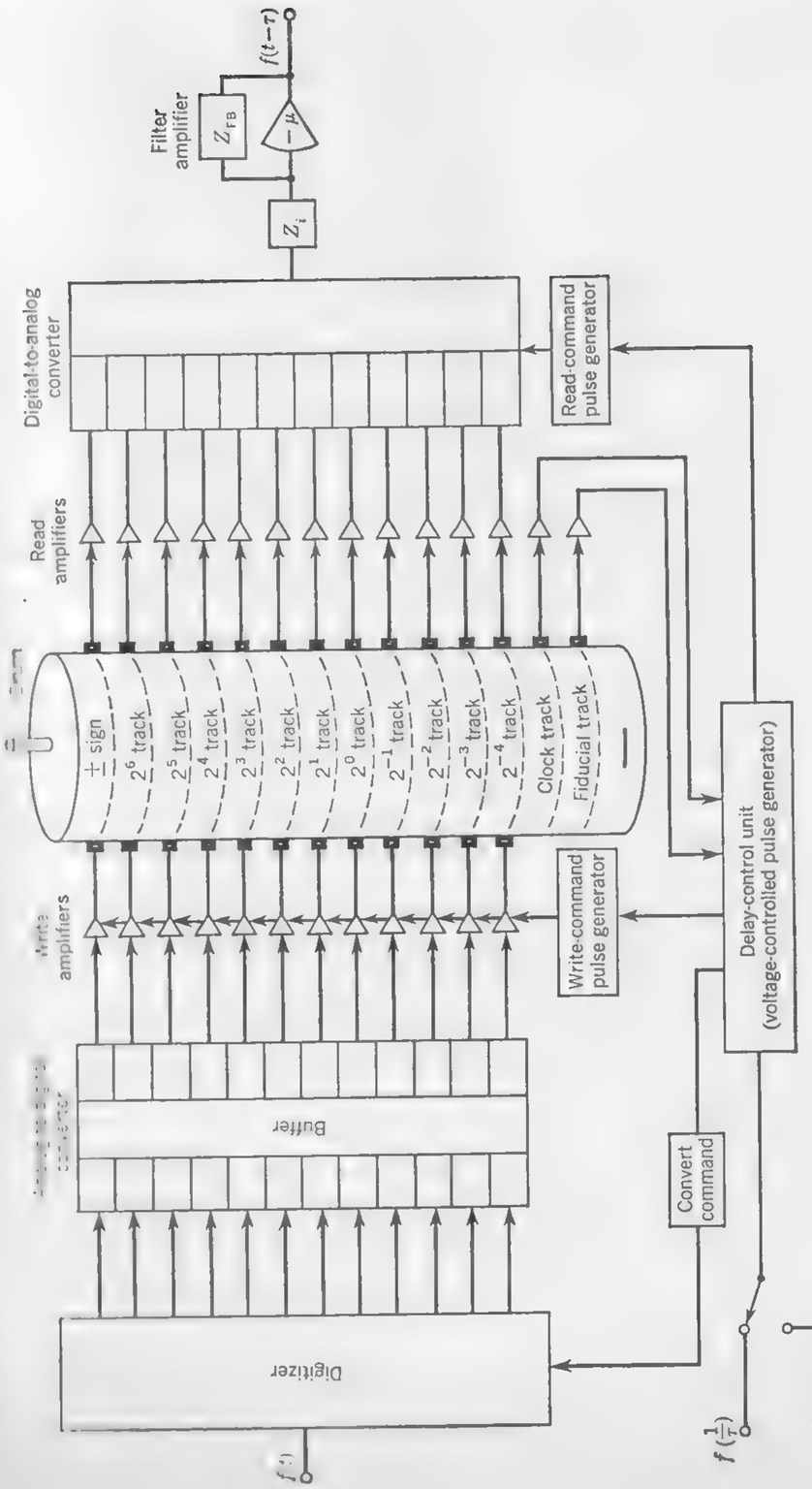


FIG. 6.1.10. Digital-drum time-delay unit. (*Simulag Group Model 1.011. Electronics Associates, Inc., Long Branch, N.J.*)

stores the binary output from the drum in its buffer register (transistorized flip-flops) upon command from a *delay-control unit*. The buffer register holds the previous binary word until commanded by the delay-control unit. The output of the digital-to-analog converter is in staircase form and is filtered by a transistorized operational amplifier (filter amplifier).

The most complex functions in the Magnetic Drum Simulag Group 1.011 are those performed by the delay-control unit. Basically, this device converts an input analog signal $1/\tau$ to a chain of pulses whose period is directly proportional to the input signal, i.e., inversely proportional to the desired delay τ. Once the sampling periods are generated, two operations are then under control—digitizing the incoming analog signal $f(t)$ and restoring the digitized signal $f(t-\tau)$ to analog form. Following each pulse from the delay-control unit a new quantized digital sample is stored at adjacent locations on the drum. Also, every pulse from the delay-control unit precesses the drum address by one count; so that after a predetermined number of counts, the first sample is sent through the read amplifiers to the digital-to-analog converter. The summation of the number of samples times the period determines the delay τ. Synchronization between the sampling event and the correct drum address is prompted by the pulses from the clock track.

Since the drum has a maximum capacity of 1,024 words, the pulse rate and resolution [samples per cycle of $f(t)$ signal] must be varied according to the delay period τ. The maximum rate is limited by the drum speed of 120 rps and the resolution by the frequency of the incoming signal $f(t)$. If 10 samples are taken for every 360 deg of input a maximum frequency of $f(t)$ can be determined as $120 \div 10 = 12$ cps. A manually operated *range-control switch* permits delays from 0.01 to 10,000 sec. Table 6.1.1 lists important characteristics of this time-delay unit.

6.1.17. Use of Servo Tables. Two-carriage servo recording tables can be used to record an input function $f(t)$ with conductive ink and to read $f(t-\tau)$ with a curve-follower attachment after a suitable time delay τ.

References

1. Truxal, John G.: *Control Engineers' Handbook*, chap. 16, McGraw-Hill Book Company, Inc., New York, 1958.
2. Tsien, H. S.: *Engineering Cybernetics*, chap. 8, McGraw-Hill Book Company, Inc., New York, 1954.
3. Gallagher, John M., Jr., and William W. Seiffert: Simulation Techniques for Nuclear Reactor Studies, speech at MIT, September, 1957.
4. Wood, John: Controlling the Pure Delay Plant, *Instruments and Automation*, September, 1957.
5. Cook, J. W.: An Automatic Gauge Control System for Tandem Cold Mills, speech at Westinghouse Electric Co., September, 1957.
6. Truxal, John G.: *Automatic Feedback Control System Synthesis*, chap. 9, McGraw-Hill Book Company, Inc., New York, 1955.
7. Stubbs, G. S., and C. H. Single: Westinghouse Atomic Power Division, Pittsburgh, *Tech. Rept.* 38, May, 1952.
8. Rogers, Allan E.: *Princeton Computation Center Rept.* 101, Electronic Associates, Inc., Long Branch, N.J., August, 1957.
9. Single, Charles H.: Time Delay Circuits for Analog Computers, Berkeley Division of Beckman Instruments, Inc., Richmond, Calif., March, 1956. A brief version of this report was reprinted in *Control Eng.*, October, 1956.
10. Johnson, Clarence L.: *Analog Computer Techniques*, McGraw-Hill Book Company, Inc., New York, 1956.
11. Smith. G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill Book Company, Inc., New York, 1959.
12. Warfield, J. N., and L. E. Weaver: Cut-product Approximants for Time-delay in Electronic Analog Computers, IRE Regional Conference Paper, Dallas, 1958.
13. Warfield, J. N.: *Introduction to Electronic Analog Computers*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1959.
14. Taskett, D. A.: Transportation Delay and Its Simulation with Analog Computers, *Proc. ISA*, May, 1958.
15. Stone, R. S., and R. A. Dandl: A Variable Function Delay for Analog Computers, *IRE Trans. PGEC*, September, 1957.

16. Kozak, W. S.: An Analog Memory, paper presented at WESCON, 1958: Canadian Westinghouse Co., Ltd., Electronics Division, Malton, Ont., 1958.
17. Landauer, Paul J.: *Princeton Computation Center Rept.* 130, Electronic Associates, Inc., Long Branch, N.J., July, 1958.
18. Jonssen, J. M. L.: Discontinuous Low Frequency Delay Line with Continuously Variable Delay, *Nature*, **169**:148 (Jan. 26, 1952).
19. Paynter, H. M. (ed.): *A Palimpsest on the Electronic Analog Art*, p. 163, George A. Philbrick Researches, Inc., 1955.
20. Truxal, John G.: *Automatic Feedback Control System Synthesis*, p. 505, McGraw-Hill Book Company, Inc., New York, 1955.
21. Kennedy, J. D., and M. M. Brenner: Dead Time Simulation for Electronic Analog Computers, paper presented at the 1957 National Simulation Council, Electronic Associates, Inc., Long Branch, N.J., 1958.
22. Kohr, R. H.: Simulation of Automobile Suspension and Steering Systems, paper presented at Midwestern Simulation Council in September, 1958, General Motors, Detroit, Mich., 1958.
23. Sampson, D. K., and H. L. Daniel: Magnetic Drum Provides Analog Time Delay, *Electronics*, **32**(6):44–47 (Feb. 6, 1959).
24. Hofman, C. D., and H. L. Pike: Analog Time Delay System, *Proc. Western Joint Computer Conference*, San Francisco, May, 1960.

PART 2: REPETITIVE ANALOG COMPUTERS

By PAUL E. RUSSELL

INTRODUCTION

6.2.1. Repetitive Analog Computers. In contrast to the related "slow" d-c differential-equation solvers, the machine variables (voltages) in a *repetitive analog computer* are reset periodically between 5 and 60 times a second to permit display of differential-equation solutions on a cathode-ray oscillograph. For most control-system problems, this requires an accelerated time scale of 1:1,000 or more (Sec. 5.1).

Inasmuch as system design (parameter optimization) by analog techniques usually requires repeated trials, the repetitive-computer display permits the operator to observe the effects of parameter changes on the solution almost instantaneously and to converge rapidly on a desired optimum solution; he need not wait a relatively long period for each solution and compare runs, as with "slow" analog computers. One can easily display parametric plots of one machine variable against another by connecting the variables to the horizontal and vertical deflection-amplifier inputs of the oscilloscope. Permanent records can be obtained with any oscilloscope camera, or by the sampling technique described in Sec. 6.3.

6.2.2. Repetitive vs. "Slow" Electronic Analog Computers. The high-speed repetitive analog computer affords savings in computing time and is usually smaller and requires less power. The first cost of a repetitive machine is often less than that of a "slow" computer having a comparable capacity. But accuracy must be sacrificed: since the working frequencies of repetitive machines exceed those of "slow" analog computers by a factor of 10^3 or 10^4, frequency-response limitations of computing elements become critical. It is essentially impossible to obtain operational-amplifier loop gains comparable with those attained in "slow" computers, and the nonlinear computing elements present even greater problems.

The best repetitive analog computers have component accuracies little better than ½ per cent of full scale, depending on frequency. The accuracy of particular solutions depends on the problem and scaling; 3 per cent of full scale is a reasonable expectation.

6.2.3. Applications. The particular advantages of high computing speed and low cost, obtained at the expense of accuracy, make the repetitive computer well suited to solution of problems where many trial solutions are necessary. In control-system design the early exploratory studies can be made quickly on this type of computer. Variation of coefficients and other parameters will quickly determine the regions of

Table 6.2.1. Special Repetitive-computer Components by George A. Philbrick Researches, Inc.*

Component	Designation	Relationship established
Adder	K3-A	$X_0 = X_1 + X_2 + X_3 + X_4$
Limiter	K3-B	$X_0 = X_1 \ (-B \le X_1 \le B)$ $X_0 = B \quad (X_1 \ge B)$ $X_0 = -B \quad (X_1 \le -B)$
Coefficient component	K3-C	$X_0 = K X_1 \ (0 \le K \le 100)$
Differentiating component	K3-D	$X_0 = \tau_0 \frac{dX_1}{dt}$ τ_0 adjustable from 0 to 40 msec or can be set by external capacitor
Augmenting differentiator	K3-E	$X_0 = X_1 + \tau_0 \frac{dX_1}{dt}$ τ_0 adjustable from 0 to 40 msec or can be set by external capacitor
Backlash component	K3-H	$X_1 - \frac{H}{2} \le X_0 \le X_1 + \frac{H}{2}$ H adjustable from 0 to 10 volts
Integrating component	K3-J	$X_0 = \frac{1}{\tau_0} \int X_1 \, dt$ (τ_0 = 0.0004, 0.004, 0.04, or 1.0 sec)
Augmenting integrator	K3-K	$X_0 = X_1 + \frac{1}{\tau_0} \int X_1 \, dt$ τ_0 adjustable from 0 to 40 msec or can be set by external capacitor
Unit-lag component	K3-L	$X_0 = X_1 - \tau_0 \frac{dX_0}{dt}$ τ_0 adjustable from 0 to 40 msec or can be set by external capacitor
Squaring component	K3-S	$X_0 = AX_1^2$
Square-root component	K3-T	$X_0 = A\sqrt{X_1}$
Absolute-value component	K3-V	$X = A\lvert X_1 \rvert$
Inert-zone component	K3-Z	$2X_0 = 2X_1 + \lvert X_1 - Z_1 \rvert - \lvert X_1 + Z_2 \rvert$
Arbitrary function generator	FFR	Diode type with quadratic rounding and self-contained power supply
Multiplier/divider (2 per unit)	MU/DV	$X_0 = \frac{X_1 X_2}{25}$ or $X_0 = 25 \frac{X_1}{X_2}$
Function of two variables	F2V	$X_0 = f(X_1, X_2)$
Central-signal component	CS-2	Provides repetitive ramp, sweep, step, and clamp. Repetition rate, 0.02 to 20 cps
Central-response component	CRM and CR	Provides a raster for oscilloscope display
Regulated power supplies	R-100A, RR-200, SR-400	±300 volts at 100 ma, 200 ma, 400 ma; 6.3 volts a-c

* Manufacturer's specifications; most units incorporate an internal output-phase inverter so that X_0 as well as $-X_0$ is available.

Table 6.2.2. Special Repetitive-computer Components by GPS Instrument Company, Inc.*

Component	Designation	Relationship established
Integrator	Part of BCU 600D	$X_0 = \int (X_1 + X_2) \, d\tau$, $-X_0$ also available
Summing amplifier	Part of BCU 600D	$X_0 = X_1 + 5X_2 + 5X_3 + 10X_4$
Coefficient potentiometer	Part of BCU 600D	$X_0 = KX_1$, K adjustable from 0 to 1
Limiter unit	LM150D	$X_0 = -(X_1 + X_2)$ $-B_1 \le X_1 + X_2 \le B_2$ $X_0 = -B_2 \quad X_1 + X_2 \ge B_2$ $X_0 = B_1 \quad X_1 + X_2 \le -B_1$
First-order lead-lag unit	FOL 350D	$\frac{X_0}{X_1 + X_2} = \frac{1}{As + 1}$ or $\frac{X_0}{X_1 + X_2} = \frac{As}{As + 1}$ A is adjustable from 0.01τ to 20τ
First-order lead-lag unit	FOLD 325C	Same as FOL 350D but with a differentiator also
Second-order lag unit	SOL 400C	$\frac{X_0}{X_1 + 5X_2 + 5X_3 + 10X_4} = \frac{A_0}{s^2 + A_1 s + A_0}$ A_0 variable from 1 to 990. A_1 variable to provide damping ratios less than 0.1 up
Function generator	FG 700D	$X_0 = f(X_1)$, diode type
Function-generator summing amplifier	FGSA800D	$X_0 = F(X_1) + X_2 + X_3 + X_4 + X_5 + 5X_6 + 5X_7 + 10X_8$
Sine-cosine unit	SC 900C	$X_0 = 25 \sin 10X_1$ or $X_0 = 25 \cos 10X_1$, the angle in degrees limited to ±200 deg
Multiplier/divider	MU 500E	$X_0 = \frac{X_1 X_2}{50}$ or $X_0 = 50 \frac{X_1}{X_2}$
Noise generator	NG 1000F	Gaussian noise output shaped by adjustable first-order lag
Probability-distribution analyzer	PDA 675F	Amplitude-distribution analyzer (see Sec. 5.9)
Master generator	MG100	Provides repetitive step, pulse, sweep, and clamp. Repetition rate 10 to 50 cps
Power unit	PR200	Regulated. 1,500 ma at + and −300 volts. 200 ma at 100 volts. 240 amp at 6.3 volts a-c

* Manufacturer's specifications. A *dead-zone/hysteresis unit* and an *electronic-relay* unit are also available.

instability and the effectiveness of possible feedback systems. It is most practical to determine approximate transfer functions of a physical system from a set of system responses by synthesizing the approximate system and adjusting the parameters of the repetitive computing system until the synthesized response agrees with that of the physical system. Likewise, the approximate solution of certain boundary-value problems and the computation of Fourier integrals, superposition integrals, and correlation functions are very reasonable applications of repetitive computers,[1,2,3] and special equipment permits rapid computation of random-process statistics averaged over a large number of computer runs ("Monte Carlo" technique, see also Sec. 6.3). Another possible application is to adaptive control systems. Suitable automatic-programming techniques (Secs. 4.2 and 5.11) could continuously optimize system parameters in the accelerated time frame of the analog, so that the adaptive system could refer to the analog periodic at intervals for configuration information.

SPECIAL COMPONENTS

6.2.4. Survey. Many computing elements used with "slow" analog computers are adaptable to repetitive computers. The repetitive-computer installation will commonly consist of operational amplifiers, potentiometers, and passive linear circuit elements for simulation of linear operations; diodes, function generators, and electronic

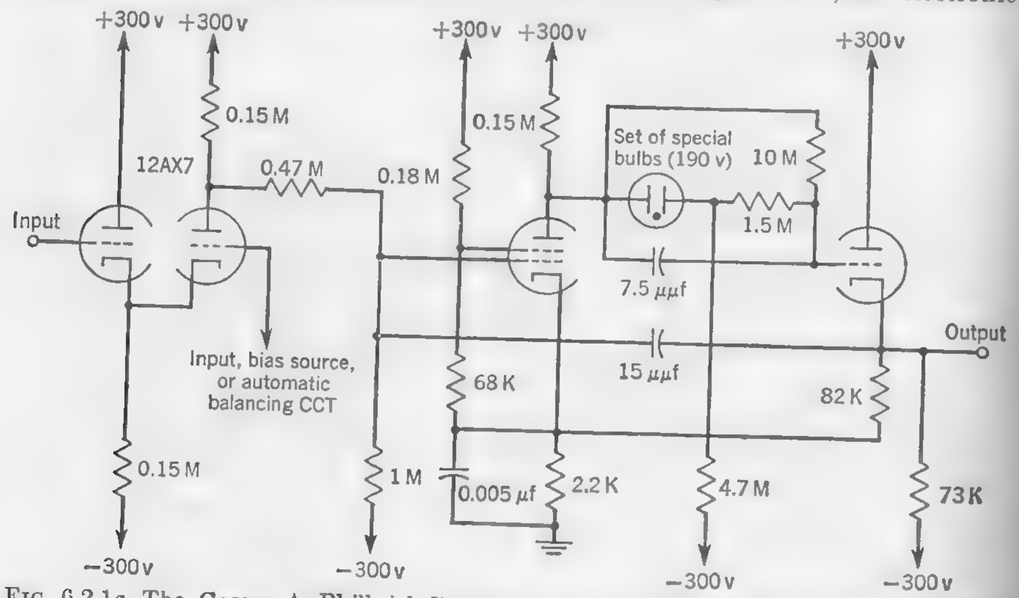


FIG. 6.2.1a. The George A. Philbrick Researches, Inc., K2-X operational amplifier plug-in unit. D-C gain, 30,000; output range, ±100 volts at 3 ma with external booster resistor, drift rate = ±5 mv per day; response, 1 μsec rise time with bandwidth over 250 kc as a unity-gain follower (output shorted to input No. 1). A compatible chopper stabilizer (K2-P) is available as a separate plug-in unit. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

multipliers for simulation of nonlinear operations; a control unit; a power supply, a display unit; and patchcords. Features of these components unique to their application in repetitive computers are described below.

6.2.5. Amplifiers. Considerations especially pertinent to the design of operational amplifiers for repetitive computers are: (1) Slow d-c amplifier drift is less important, since solution times are of the order of milliseconds rather than seconds or minutes as in "slow" computers. (2) A pure d-c level can be distinguished from a constant component of a machine variable when displayed on an oscilloscope so that, in principle, it is not necessary to stabilize every amplifier in an installation. (3) Any stabilization network or other design technique which adversely affects the high-frequency response

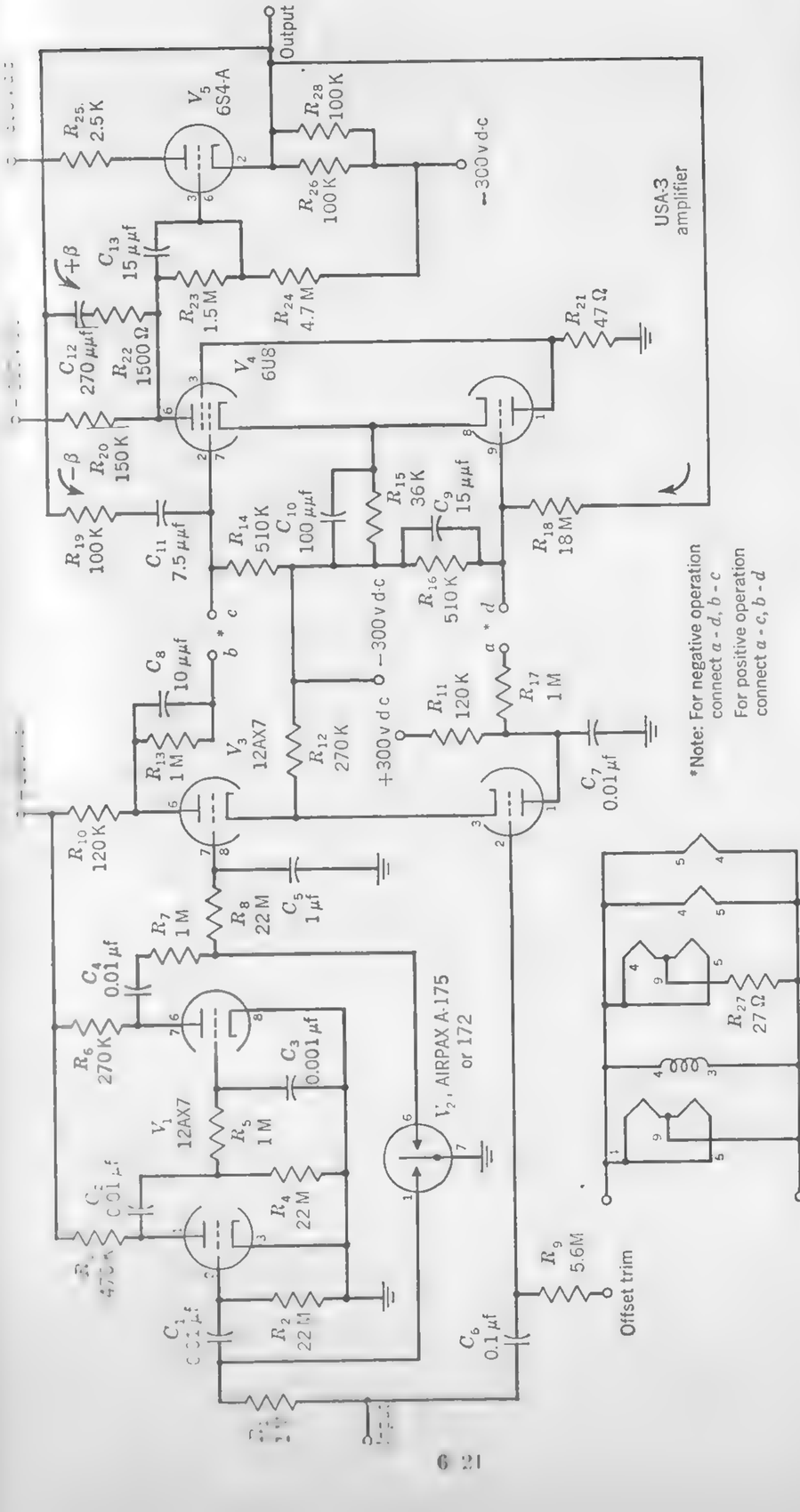


FIG. 6.2.1b. The George A. Philbrick Researches, Inc., USA-3 stabilized operational amplifier. D-C gain, 10,000,000; output range, ±100 volts at 8 ma; drift rate = ±50 μv per day; response, down 3 db at 10 kc as an amplifier with gain of 100.

of the amplifier will be detrimental. (4) Physical limitations on attainable gain-bandwidth products limit the useful amplifier gain at computing frequencies, nor do the accuracies commonly obtainable in the nonlinear components and in the display systems of repetitive computers warrant the use of amplifier gains as high as those employed with "slow" analog computers. Gains of the order of 1,000 to 10,000 are commonly in use. (5) Response times of reset circuits must be much faster than in "slow" computers.

The principles of operational-amplifier design are described in Secs. 2.3 and 7.1. Two circuits useful in repetitive computers are shown in Fig. 6.2.1. These are d-c

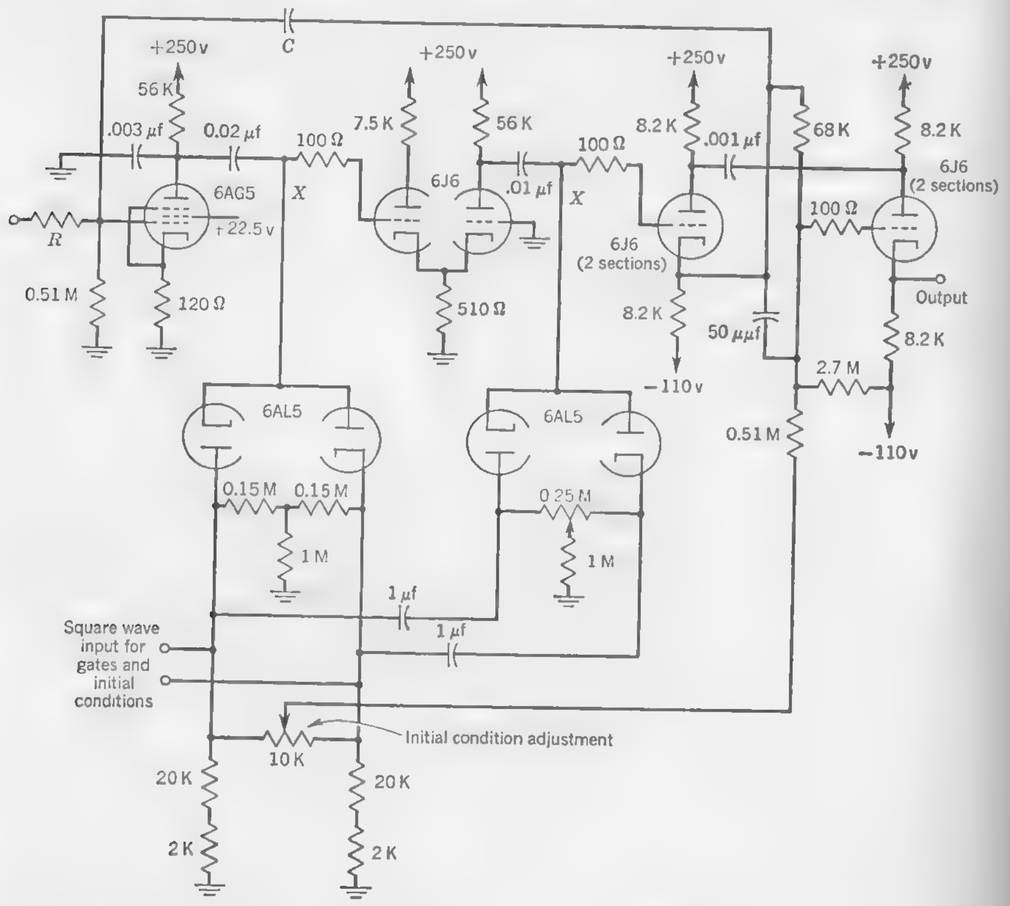


Fig. 6.2.2. Circuit diagram of an integrator from the MIT repetitive computer. The open-loop gain is 1,500, and the open-loop half-power points are at 0.16 cps and 100 kc. Output range is ±20 volts at about 1 ma with less than 3 per cent distortion. (*From G. A. Korn and T. M. Korn, Electronic Analog Computers, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956.*)

amplifiers and require well-regulated power supplies to avoid common power-supply impedance as well as excessive drift in the reference levels of the solutions. A rather different approach is indicated in Fig. 6.2.2. This a-c amplifier was developed under the direction of Dr. A. B. MacNee of the Research Laboratory of Electronics of the Massachusetts Institute of Technology.[1] In this type of amplifier, power-supply regulation is not an essential requirement, and operation from a single power supply is possible. Interaction between computing elements through the power supply can be eliminated by the use of conventional decoupling filters.

6.2.6. Reset Circuits. The techniques useful in resetting integrators and similar components to their proper initial conditions differ from those used in "slow" com-

puters. Relay resetting is used only with the slowest repetition rates (repetitive operation of "slow" computers, Sec. 4.1). If the initial conditions are zero, and the computer responses are well damped and die out during the computing period, no provision for reset is required. Otherwise, provision must be made to reset the integrators at the beginning of each repetition period. One approach is shown in Fig. 6.2.3*a*. In this arrangement, periodic positive reset pulses from a central control unit clamp the integrator output to zero by shorting the capacitor through the double triode, which

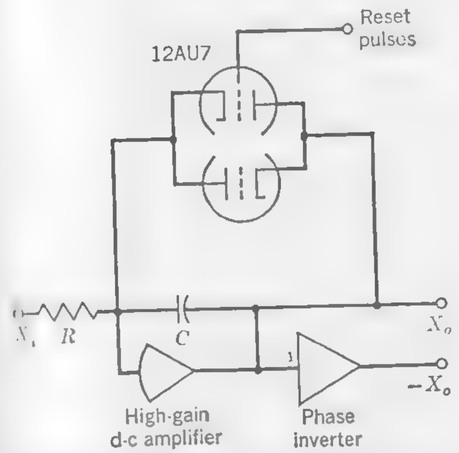


Fig. 6.2.3*a*. Electronic resetting circuit of an early Philbrick integrator. Initial conditions other than zero must be introduced by an external summing amplifier.

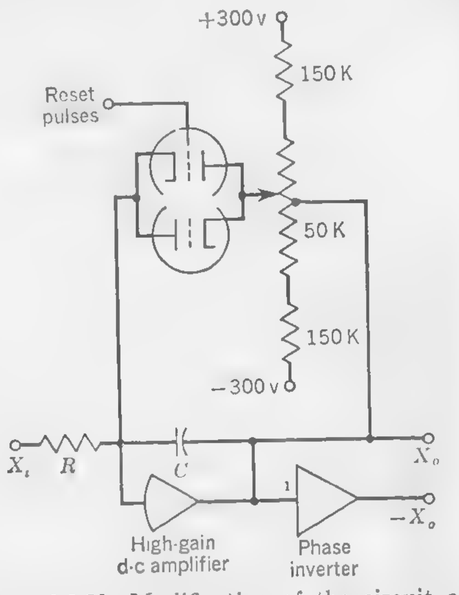


Fig. 6.2.3*b*. Modification of the circuit of Fig. 6.2.3*a* developed at the University of Arizona. Initial conditions different from zero can be set directly (from Ref. 2).

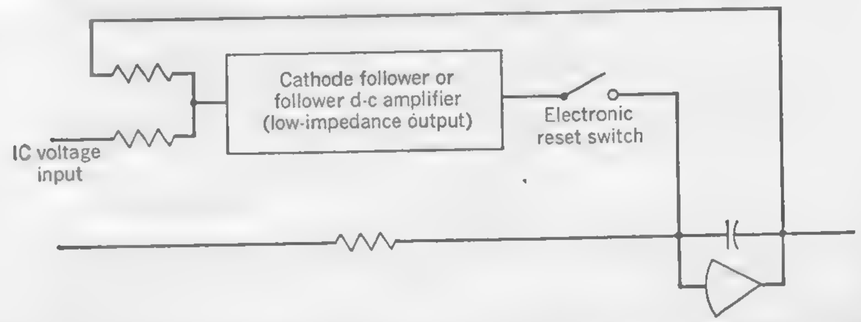


Fig. 6.2.3*c*. An improved resetting circuit permitting direct setting of initial conditions. S_1 is, preferably, a diode-bridge switch (Fig. 6.3.2).

is otherwise biased beyond cutoff. This clamping arrangement requires that nonzero initial conditions be inserted as d-c inputs in a summing circuit following the integrator. Similarly, in the circuit of Fig. 6.2.2, positive and negative gating pulses permit the clamping diodes to conduct and reduce the output voltage of each amplifier stage to zero. The initial-condition voltage is not used to charge the integrating capacitor but is added to the integrator output voltage in the last amplifier stage.

Figure 6.2.3*b* represents a considerable improvement, since an initial-condition voltage other than zero can be inserted without an external summing circuit;[2] this is a very useful modification for early-model Philbrick integrators. The circuit of Fig.

6.2.3*c* is similar but disables the integrator faster and more positively during the reset interval.[3,4]

Diode-bridge and transistor switches (Fig. 6.3.2) can serve as electronic switches in Fig. 6.2.3*c* with improved results. The six-diode switch[5] shown in Fig. 6.3.2 as a sample-hold resetting switch is probably the most accurate resetting switch available.[6] Another interesting reset circuit is used in Philbrick's newer K3-J integrating units

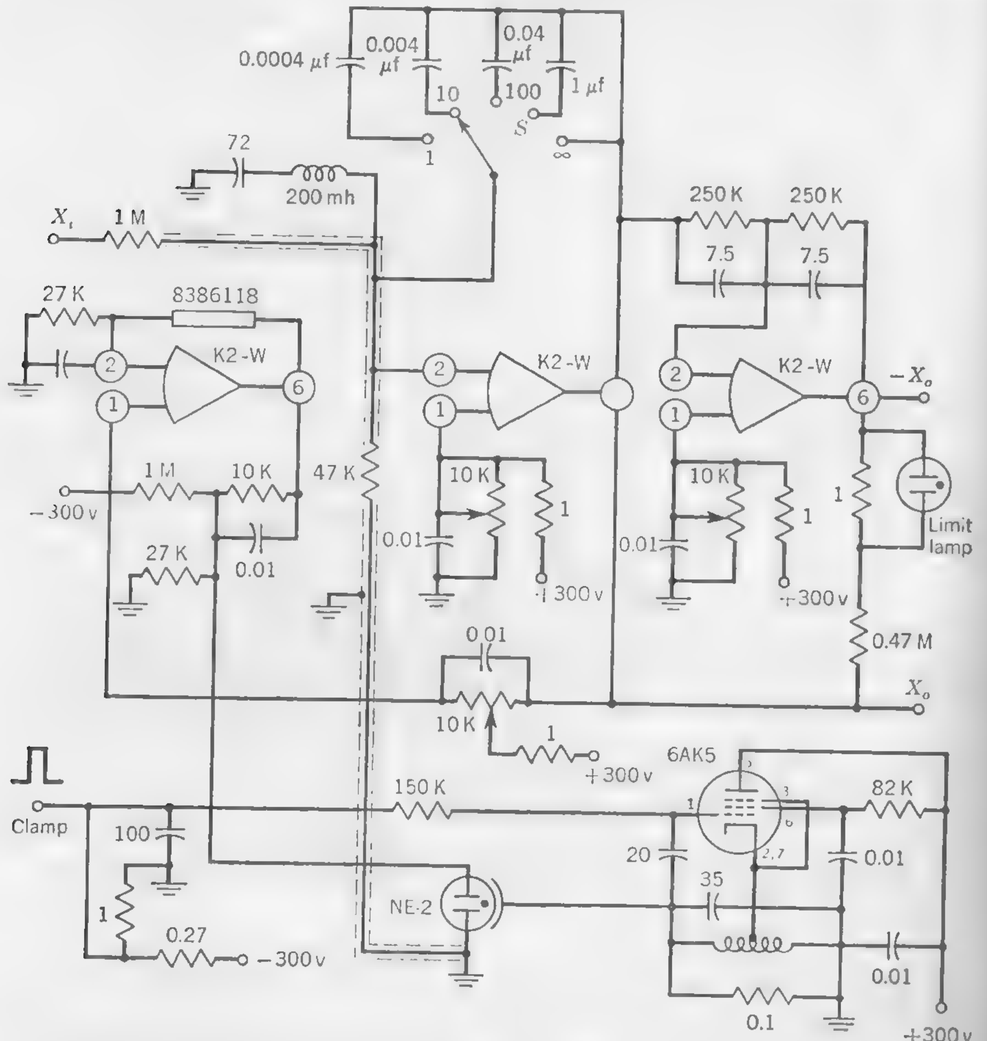


FIG. 6.2.3*d*. This Philbrick Model K3-J integrator uses a neon bulb ionized by a pulsed oscillator to short-circuit the integrating capacitor. Initial conditions other than zero are introduced by an external summing amplifier.

(Fig. 6.2.3*d*). In this system the reset pulse triggers an oscillator, which in turn ionizes the gas in a neon tube to reduce the initial-condition voltage to zero. No provision is made for setting an initial-condition voltage other than zero except through an external summing amplifier.

Figure 6.2.4 shows an entirely different accurate reset circuit which could substitute for the relay reset circuit of a "slow" integrator.[2]

6.2.7. Control Unit and Displays. The control unit for a repetitive computer provides the repetitive forcing functions, steps, ramps, clamps, and either synchronization for oscilloscope sweeps or actual sweep voltages.[6 to 9] Displays may take the

form of voltages repetitively displayed against time (sweep applied to horizontal oscilloscope plates), two voltages displayed against one another (phase-plane plots), or multiple traces displayed by brightening of a television-type raster. In each case the control unit can generate calibrating pulses corresponding to preset voltages to provide

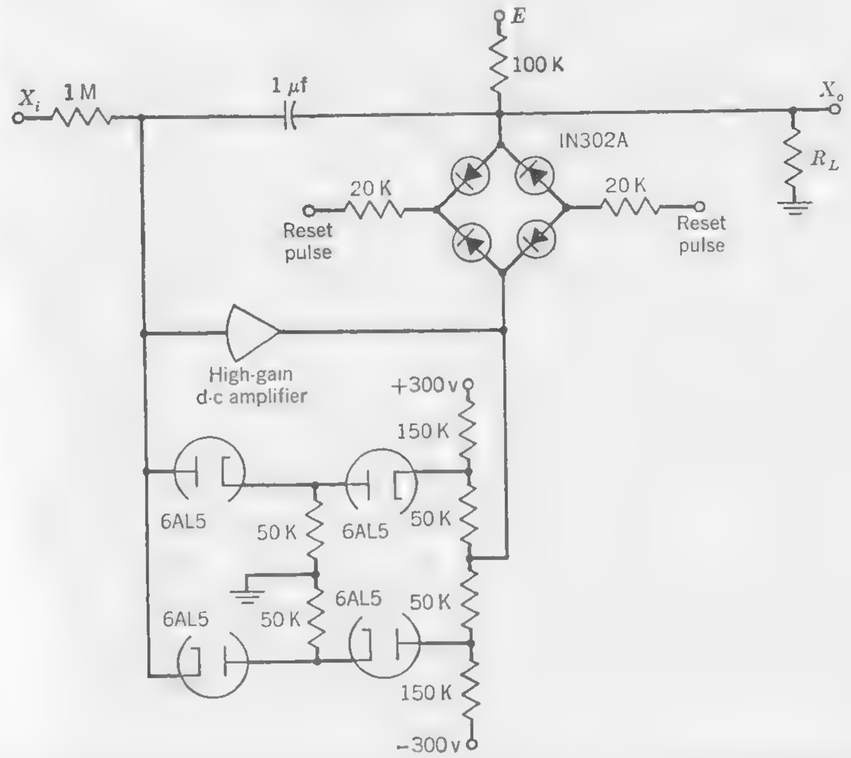


FIG. 6.2.4. This electronic resetting circuit charges the integrating capacitor directly rather than through the integrating amplifier. A special low-leakage feedback limiter circuit holds the summing point at ground potential in RESET.

for quick quantitative interpretation of the display, and to make the display independent of deflection linearity.[8]

References

1. MacNee, A. B.: An Electronic Differential Analyzer, *Proc. IRE*, **37**:1315 (1949).
2. Koerner, H., and G. A. Korn: New Operational-amplifier Techniques for Precision Switching and Function Generation, *Electronics*, Nov. 6, 1959.
3. Diamantides, N. D.: A Multipurpose Electronic Switch for Analog Computer Simulation and Autocorrelation Applications, *IRE Trans. PGEC*, December, 1956.
4. Lofgren: Predictors in Time-shared Analog Computers, *Proc. International Analogy Computation Conference*, September, 1955.
5. Millman, J., and T. H. Puckett: Accurate Linear Bidirectional Diode Gates, *Proc. IRE*, January, 1955.
6. Korn, G. A.: Repetitive Analog Computers at the University of Arizona, *Instruments and Control Systems*, September, 1960.
7. Specification Sheet for Model CS-2 Central Signal Component, G. A. Philbrick Researches, Inc., Boston, Mass.
8. Specification Sheet for Model CRM Central Response Component, G. A. Philbrick Researches, Inc., Boston, Mass.

A large number of applications are discussed in *A Palimpsest on the Electronic Analog Art*, George A. Philbrick Researches, Inc., Boston, Mass., 1955.

PART 3: ANALOG STORAGE TECHNIQUES FOR ELECTRONIC ANALOG COMPUTERS

By GRANINO A. KORN

ANALOG STORAGE CIRCUITS AND APPLICATIONS

6.3.1. Introduction. *Sample-hold circuits* are analog storage devices which permit read-out and/or storage of the value $x(t_1)$ of a variable voltage $x(t)$ at a specified *sampling time* t_1. Such memory circuits can give an electronic analog computer much of the decision-making and program-controlling ability of a digital computer and are therefore steadily increasing in importance. Section 6.3 deals mainly with applications involving periodic sampling, which preserves the information content of suitably band-limited input voltages at the sampler output. *A number of other applications of analog storage circuits are discussed in Secs.* 4.2, 5.3, 5.4, 5.11, 6.1, 6.5, *and* 6.6.

6.3.2. Sample-hold Circuits.[1 to 5] Figure 6.3.1*a* shows a *basic capacitor storage circuit.* In more exacting applications, this is replaced by the *operational-amplifier zero-order sample-hold circuits* of Fig. 6.3.1*b* and *c*, which act as integrators in RESET during sampling (switch S closed), and as integrators in HOLD during the hold interval. Typical waveforms for periodic sampling are also shown.

For a sample-hold accuracy of 0.1 per cent the sampling time (switch S ON) must exceed the capacitor-charging time constant by a factor of 7, and the hold-circuit time constant (C times leakage resistance) should exceed the holding interval (S OFF) by a factor of 1,000.[4] The speed-up capacitor C_1 in Fig. 6.3.1*b*, effectively cancels the charging delay introduced by R and C, or the low-impedance follower circuit of Fig. 6.3.1*c* can be used to reduce the effective value of R. The charging time depends, then, essentially on the ability of the amplifier to supply current pulses to a capacitive load. The usual d-c integrator drift, leakage, and dielectric absorption are additional error sources. The simple circuit of Fig. 6.3.1*b* is probably as good as any for working frequencies below 1 kc. At higher frequencies, the input speed-up capacitor C_1 keeps the electronic switch from closing, and the circuit of Fig. 6.3.1*c* must be used.

Figure 6.3.1*d* shows how a measure of *prediction* can be added to a sample-hold circuit (*rate-aided hold circuit*).[3] Higher-order predictor circuits can be built in an analogous manner. Sample-hold circuits incorporating prediction can be extraordinarily useful in multiplexing applications (Art. 6.3.4), since they often permit the use of lower sampling rates for a given accuracy.[3] Reference 6 shows a number of more elaborate circuits for implementing higher-order holds.

In critical applications, samplers may have to be preceded by low-pass filters to prevent "aliasing" errors due to spurious high-frequency noise, but many "slow" d-c analog-computer circuits already discriminate against spurious high-frequency components.

6.3.3. Accurate Electronic Switches (see also Sec. 3.3).[7 to 10] In the earliest applications, the sampling switch S indicated in the various circuits of Fig. 6.3.1 was a relay, commutator, or cam-operated switch. These mechanical devices limit the sampling times obtainable to time intervals larger than about 1 msec, with a comparable lower limit on timing accuracy. Much faster switching rates, and frequently also cleaner switching, can be obtained with the *electronic switches* shown in Fig. 6.3.2.

The four-diode switches of Fig. 6.3.3*a* and *b* are the easiest to patch on "slow" multipurpose analog computers. The six-diode switch of Fig. 6.3.2*c* is probably the most accurate sampling-multiplexing switch.[21] Note that only one-half of each diode-bridge circuit is needed if the input voltage is known to be of one sign. The switching circuits of Fig. 6.3.2 are very useful in many other analog-computer applications, e.g., for resetting integrators in repetitive analog computers (Sec. 6.2; see also Sec. 3.3).

Depending on the application, accurately timed gate pulses (sampling pulses) to operate the electronic switches may be obtained from special multivibrators, blocking oscillators, or ring counters, which can run free or in synchronism with crystal or tuning-fork sources, or with the 60-cps line frequency. In "slow" simulation systems, sampling pulses are conveniently generated by computer-simulated multivibrators (Sec. 3.3). More sophisticated systems will derive nonperiodic sampling pulses from comparators in the analog-computer setup itself (Art. 6.3.12).

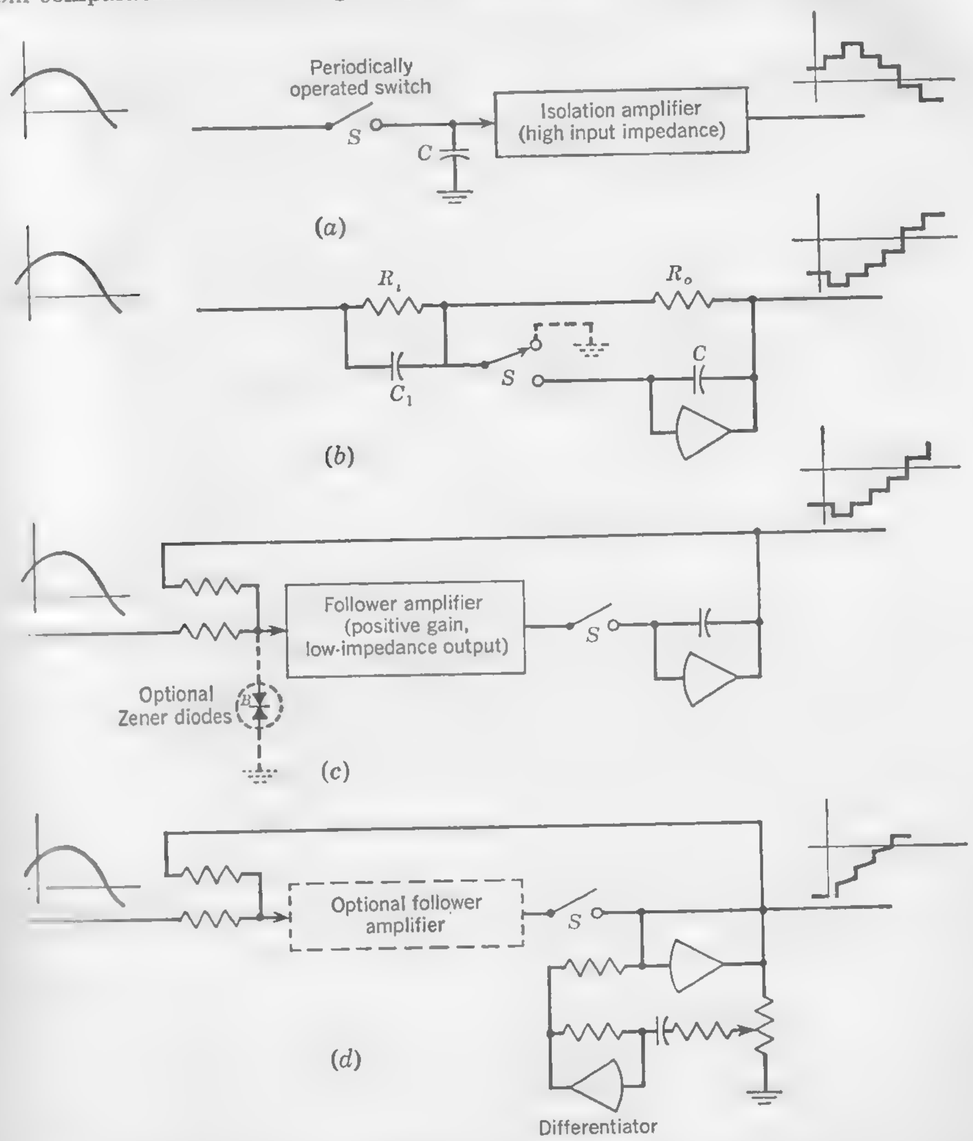


Fig. 6.3.1. Analog-computer sample-hold circuits. All but Fig. 6.3.1*d* are zero-order sample-hold circuits. The optional Zener-diode limiter indicated in dash lines in Fig. 6.3.1*c* permits the use of low-voltage follower amplifiers and electronic switches and helps to reduce switch leakage due to high signal and control voltages.

6.3.4. Time Sharing (Multiplexing) of Computing Elements. Since sample-hold circuits of the type shown in Fig. 6.3.1*b*, *c*, and *d* can be as accurate as operational amplifiers, they may be used together with mechanical or electronic multiplexing switches to *time-share* an analog-computing element among several computing circuits (Fig. 6.3.3). This is useful if

1. Multiplexing circuitry is more economical than multiple computing elements.
2. Accuracy is not affected by the bandwidth limitation inherent in the use of sampled data.

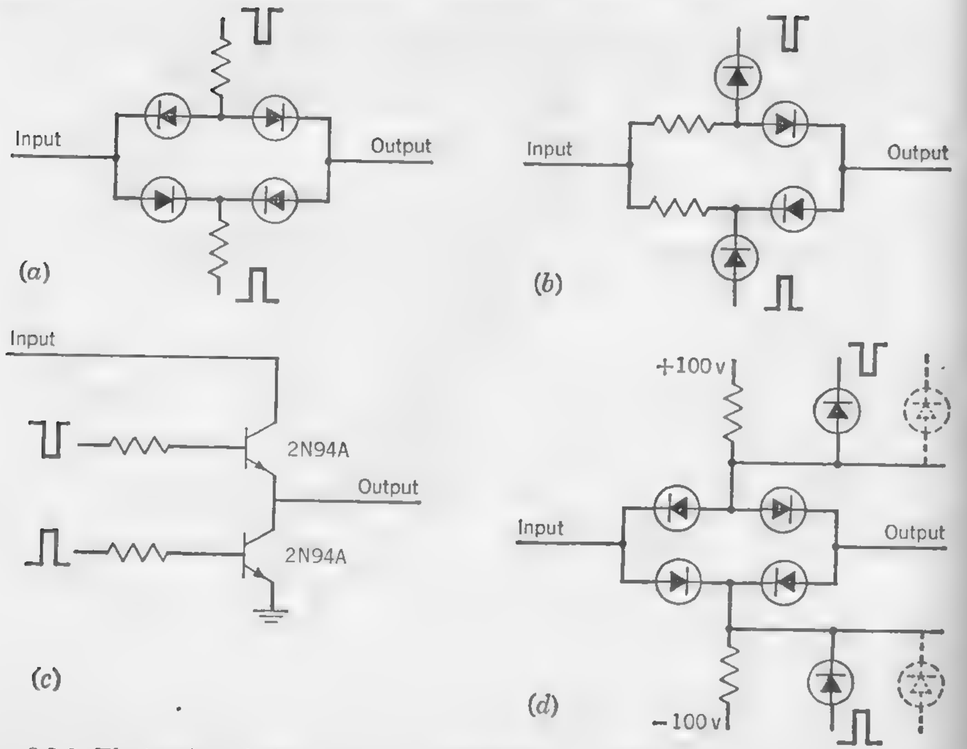


FIG. 6.3.2. Electronic switches suitable for sampling and multiplexing: four-diode bridge (*a*) and (*b*), switch using *n-p-n* transistors with symmetric characteristics (*c*), and six-diode bridge (*d*). The pulses shown turn the switches OFF. Note that the circuits of Fig. 6.3.2*b*, *c*, and *d* incorporate shunt as well as series switching action. If the input voltage does not change sign, one can omit one-half of each bridge circuit. The extra switching diodes indicated in dash lines in Fig. 6.3.2*d* permit the switch to be actuated by either one of two separate circuits.

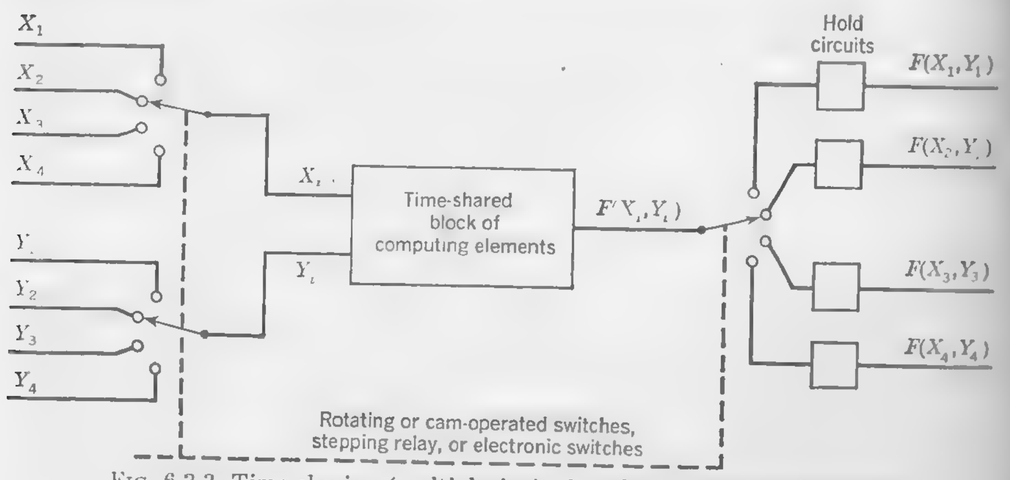


FIG. 6.3.3. Time sharing (multiplexing) of analog-computing elements.

Theoretically a signal of bandwidth B can be exactly reproduced from $2B$ samples per second; Ref. 6 discusses bandwidth-limitation effects of practical sampled-data systems in detail. If zero-order sample-hold devices imply too much of a bandwidth limitation, it may still be possible to obtain useful results with simple rate-aided sample-hold circuits (predictors, Fig. 6.3.1*d*).[3]

A significant application of multiplexers is to the set of *multiple-coordinate transformations* inherent in aircraft and missile simulation (Sec. 5.5).[3,5] These transformations involve a large number of multiplications by trigonometric functions which are expensive to implement with the required accuracy and lend themselves well to time-sharing techniques. Note that servo *potentiometer* inputs rather than *servo* inputs should be multiplexed, since fast servo slewing is undesirable.

In a typical application,[5] a multiplexer employing cam-driven microswitches effected a saving of 20 potentiometer cards (8 resolvers) and 33 amplifiers out of 28 cards and 60 amplifiers. The signal frequency was 0.1 cps, the holding time was 1 sec, and the time constant of the memory was 0.01 sec. Much faster sampling is easily achieved with electronic multiplexers, which combine well with fast and accurate (and expensive) electronic multipliers and resolvers, and also with analog-digital converters and digital function generators (Secs. 3.3 and 6.5).

6.3.5. Finite-difference Techniques. A set of sample-hold circuits set up to sample an input voltage $x(t)$ at predetermined or voltage-controlled times $t_1, t_2, \ldots$ may also be used for *numerical differentiation and integration* of $x(t)$ with respect to the time t or with respect to another voltage $y(t)$; conventional difference formulas for approximate differentiation and integration are implemented by analog addition and subtraction.[4,11] Techniques of this type can also solve *difference equations* (Sec. 6.6).

SPECIAL COMPUTERS EMPLOYING ANALOG SAMPLING TECHNIQUES

6.3.6. Stepping-relay Techniques. The sampled data simulator and computer (SADSAC) at the Franklin Institute Laboratories[12] and a simpler similar analog-computer setup at Stanford University[13] employ multiple-section stepping relays for sampled-data storage and multiplexing. Stepping relays can be operated periodically or by voltage-controlled variable-frequency oscillators. An interesting application involves multiplexing of the operational amplifiers needed for difference-differential equation solution of partial differential equations (Sec. 5.9) where a large number of stations may be required, as in nuclear-reactor heat-transfer problems. A stepping relay can also be employed to implement *transport time delays* in the manner of Art. 6.4.

Figure 6.3.4 shows how multiple-section stepping relays are used for a 12-station simulation of a counterflow heat exchanger (see also Sec. 5.7); note how the simulated primary and secondary fluids "rotate." In view of the relatively large number (30 to 100) of voltages to be stored in such problems, the simple capacitor-storage scheme of Fig. 6.3.1*a* is employed rather than an operational-amplifier integrator in HOLD. As a result, one really has an *RC* network analogy (Sec. 9.1) rather than a differential-analyzer-type computer, but the sampling and switching techniques are just as applicable to the latter.

SADSAC also incorporates "electronic switches," really time-division multipliers with multiple slave channels which can be used to introduce respectable numbers of variable coefficients into partial-differential-equation setups. Such multipliers have been added to the heat-exchanger analogy of Fig. 6.3.4 to represent variable flow characteristics and variable film coefficients.[12]

6.3.7. A Hybrid Analog-digital Computer.[13 to 18] An experimental real-time flight simulator developed by the MIT Servomechanisms Laboratory for the U.S. Naval Training Center (Port Washington, N.Y.) is a true hybrid analog-digital computer. With the aircraft flight equations (Sec. 5.5) written in the form

$$\frac{d^2x_i}{dt^2} = f_i\left(x_1, x_2, \ldots, x_n; \frac{dx_1}{dt}, \frac{dx_2}{dt}, \ldots, \frac{dx_n}{dt}; t\right) \qquad (i = 1, 2, \ldots, n)$$

the machine represents each variable x_k by an analog voltage together with a digital word indicating the correct sign as well as the scale factor or floating decimal-point position. *The analog arithmetic unit* comprises $2n$ *parallel analog integrators* but only *one analog summer, one multiplier, and one divider*, which are time-shared with the aid of electronically switched sample-hold circuits (Fig. 6.3.2*d*). A relatively simple *digital*

arithmetic unit performs the corresponding operations on the digital sign and scale words. Analog comparator circuits (Sec. 3.3) sense the magnitude of summer, multiplier, and divider output voltages and automatically change the analog-circuit gain together with the floating decimal point of the digital system, so that favorable scale factors (Sec. 5.1) are selected automatically. The digital arithmetic also communicates with the analog arithmetic by gating scale-setting resistors into operational amplifiers. The integrator outputs can be read out in analog or digital form.

A *hybrid function generator* employs digital circuitry together with analog addition, multiplication, and division. Discrete values of functions of one, two, or more variables as well as values of constant coefficients are stored in digital form on a *magnetic*

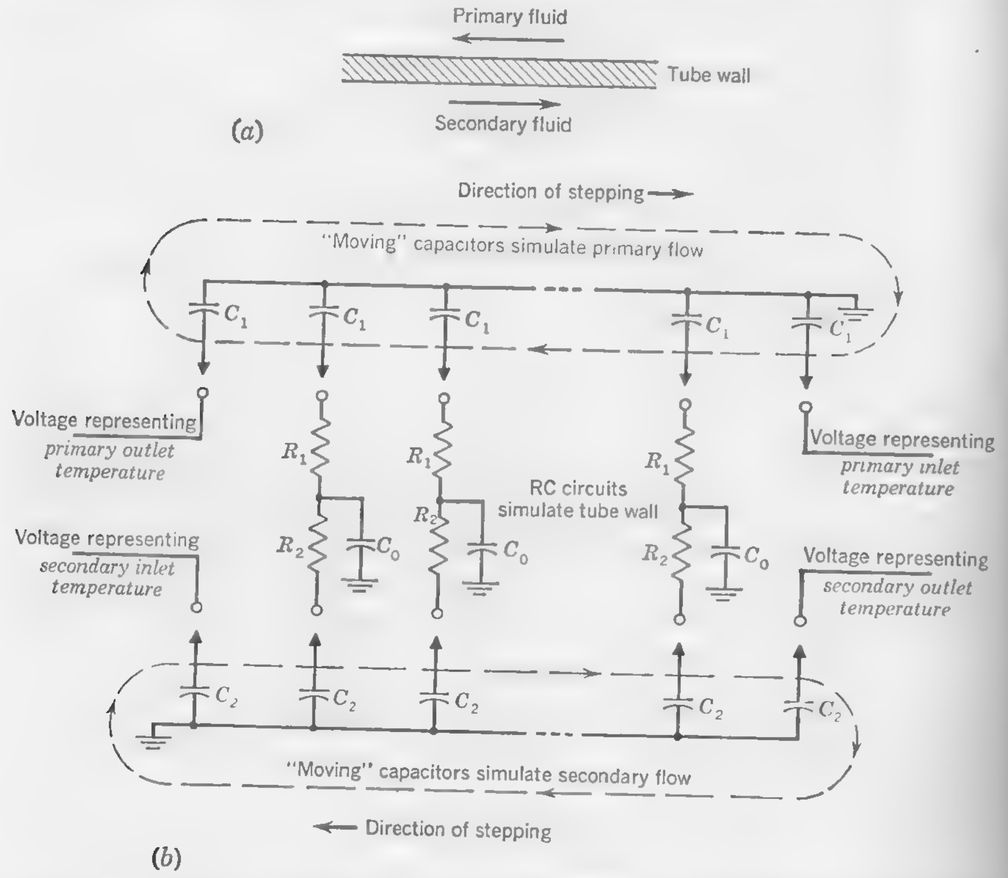


FIG. 6.3.4. Counterflow heat exchanger (*a*) and simulation with multigang stepping relay (*b*).

memory drum. These digital data are decoded and entered into the analog-digital arithmetic units, which can perform suitable interpolations.

The same drum also controls the *machine program* by a sequence of *command pulses* which control the successive operations of the arithmetic elements through a switching matrix and a multiplicity of analog and digital gates. Program, given functions, and given constant coefficients are entered on the drum prior to computation, either from a keyboard or from a stored tape. No information is entered on the drum during a computer run.

Because of the time-shared analog operations, analog voltages appear as pulse amplitudes between 0 and 50 volts, with a pulse width of about 10 sec. Component accuracy has been traded for the necessary speed of operation; thus, the analog multiplier is a simple thyrite multiplier (Sec. 3.3) with a component accuracy of 7 per cent of

full scale. This accuracy could be improved somewhat with more sophisticated computing elements.

SAMPLED-DATA TECHNIQUES FOR REPETITIVE-ANALOG-COMPUTER READ-OUT AND STATISTICAL COMPUTATIONS

6.3.8. Use of Sample-hold Circuits for Repetitive-analog-computer Read-out. With a repetitive analog computer set up to solve a set of differential equations repetitively N times per second (Sec. 6.2), a sample-hold circuit triggered to sample a solution voltage $x(t)$ t_1 msec after the start of each repetition period will yield a d-c output equal to $x(t_1)$, which can be conveniently read on a digital voltmeter. Again, one may vary the sampling time t_1 manually or by means of a controlling voltage at a rate permitting one to record $x(t_1)$ vs. t_1, or $y(t_1)$ vs. $x(t_1)$ on ordinary strip-chart recorders or servo plotting tables.

An accurate sample-hold device designed for use with a repetitive analog computer[22] employs the circuit of Fig. 6.3.1*b* and the six-diode switch of Fig. 6.3.2*c*. Accuracy should be about 0.2 per cent of full scale for sampling times and sample voltages. Associated circuitry permits one to set the sampling time t_1 either manually (multiturn potentiometer with digit-indicating dial) or by means of an external voltage. The sampling circuit may also supply a pulse which brightens or blanks the solution trace on a display oscilloscope at the time t_1 after the solution starts.

6.3.9. Repetitive-analog-computer Estimation of Ensemble Statistics in Random-process Studies.[19 to 23] If the repetitive analog computer is supplied forcing functions and/or initial-condition voltages from suitable noise generators (Sec. 3.3), each sample-hold circuit will produce a sample $x(t_1)$, $y(t_1)$, . . . of a random-process sample function (Sec. 5.9) once per repetition period. Integration of the resulting step-function voltages through a preset number n of computer runs (say 1,000 runs in 50 sec) yields estimates of the *ensemble averages* $E\{x(t_1)\}$, $E\{y(t_1)\}$, ...

Simple (and relatively slow) d-c analog operations on the sampler output voltages, followed again by integration over a preset number of computer runs, yield estimates of important ensemble statistics like $E\{x^2(t_1)\}$, $E\{[x(t_1) - y(t_2)]^2\}$ (*mean-square delay error*, Sec. 5.9), $E\{x(t_1)x(t_2)\}$, $E\{x(t_1)y(t_2)\}$ (*ensemble correlation functions*).

If a sampler output voltage $x(t_1)$ is applied to a comparator amplifier set to trigger for $x(t_1) \leq X_1$ (Sec. 3.3), a simple decade counter counting the comparator output pulses permits direct digital read-out of estimates for n Prob $[x(t_1) \leq X_1]$, so that the estimated *probability distribution function* of $x(t_1)$ is easily plotted. A more sophisticated scheme substitutes the slicer circuit of Fig. 5.9.13 for the comparator to read estimates of n Prob $[X_1 - \Delta x/2 < x(t_1) \leq X_1 + \Delta x/2]$, and thus also of *probability densities*.

Note that these sampling methods are in no sense restricted to stationary random processes but apply to nonstationary random processes as well.

6.3.10. Computation of Time Averages for Stationary Random Processes. Essentially the same techniques permit estimation of *time averages* for stationary random processes. It is only necessary to sample the random-process voltage at sampling times $t_1, t_2, \ldots, t_n$ spaced sufficiently far apart so that the samples $x(t_1)$, $x(t_2)$, . . . are at least approximately statistically independent, and to compute averages in the manner of Art. 6.3.9. Periodic samples should not be taken at rates harmonically related to the line frequency.

This method, which also lends itself well to combined analog-digital computation, is frequently the best method for computing *time-correlation functions*, since no compli-cated time-delay devices are required; the sample-hold circuits themselves supply the time delay.

To compute a correlation-function estimate

$$\frac{1}{n}\sum_{k=1}^{n} x(t_k)y(t_k + \tau) \approx R_{xy}(\tau)$$

one can sample $x(t)$ at $t_1, t_2, \ldots$ and $y(t)$ at $t_1 + \tau, t_2 + \tau, \ldots$ and multiply corresponding samples. Instead, one can sample $x(t)$ at $t_1, t_2, \ldots$, multiply the sampler output by $y(t)$ to obtain successively $x(t_1)y(t_1), x(t_2)y(t_2), \ldots$, and sample the multiplier output at $t_1 + \tau, t_2 + \tau, \ldots$ to obtain the desired products $x(t_1)y(t_1 + \tau)$, $x(t_2)y(t_2 + \tau), \ldots$.

6.3.11. An Advantage of Sampled-data Techniques for Statistical Computations. The sampled-data techniques described in Arts. 6.3.9 and 6.3.10 have, where applicable, another significant advantage. The different samples of $x(t)$ used to compute estimates of ensemble or time averages are, in either case, statistically independent random variables with identical probability distributions. Hence statistical tests and confidence intervals involving such sampled-data estimates can be derived by the well-established methods of classical random-sample statistics. This is not true for, say, correlation-function estimates calculated by direct integration over time in the manner of Figs. 5.9.1 to 5.9.3.

AUTOMATIC SEQUENTIAL COMPUTATION

6.3.12. Systems and Components. In an attempt to extend the advantages of automatic sequential operation with analog storage to existing multipurpose analog computers, several manufacturers have brought the coil terminals of the integrator control relays out to the patchbay to permit individual control of integrators. In this manner, some integrators can be placed in the HOLD condition to serve as analog storage units, one part of the computer can be in RESET while another part is in HOLD or OPERATE, etc. (Secs. 4.1 and 4.2). More recently, several computer manufacturers have introduced complete accessory systems permitting electronic switching of integrators for use as analog storage units; to exploit the possibilities of automatic sequential operation more fully, such accessory kits also permit modification of integrators to operate at greatly increased computing speeds (repetitive operation, see also Sec. 6.2). Systems of this type were pioneered by Computer Systems, Inc., with their DYSTAC system (for DYnamic STorage Analog Computer), which may be regarded as a typical example.

An accessory automatic programming system will comprise

1. *Storage capacitors* and *electronic switches* of the type shown in Fig. 6.3.2 for high-speed control of selected integrators, which may now serve as analog storage units or as ordinary integrators capable of operating at repetitive-computer speeds
2. *Timing circuits* to operate and reset the electronically controlled integrators periodically
3. *Electronic comparators* to switch analog storage units or integrators into OPERATE, HOLD, or RESET when the sum of the comparator input voltages changes sign.

Figure 6.3.5 shows a typical pair of analog storage units (memory pair) controlled by an electronic comparator. Normally, integrator 1 is in RESET and tracks the input voltage $X = X(t)$ applied to its initial-condition (IC) terminal; integrator 2 is in HOLD and stores an output voltage obtained during the last cycle of operation. When the sum of the comparator input voltages exceeds zero, the condition of the two integrators is reversed by electronic switching, so that integrator 1 now stores the last value of $-X$ obtained at the instant of switching; integrator 2 is now in the RESET condition and, therefore, reproduces X at its output terminal during the next cycle of operation. The comparator can also be actuated periodically by means of periodically applied voltages. Automatically controlled solutions may be *read out* by any of the techniques discussed in connection with conventional repetitive analog computers in Sec. 6.2; frequently, the output voltages of the storage integrators will be of principal interest. These are easily read with digital read-out equipment.

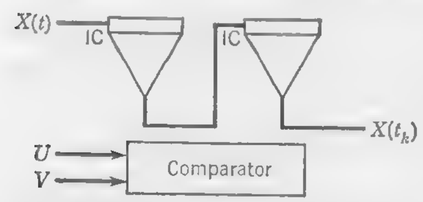


FIG. 6.3.5. Analog-memory pair. Integrator 1 is in RESET when integrator 2 is in HOLD, and vice versa; the situation is reversed whenever the sum of the comparator input voltages U and V changes sign.

6.3.13. Applications. If the memory pair of Fig. 6.3.5 is operated periodically with an input voltage $X(t)$, the output during each time interval Δt will be $X(k\,\Delta t)$ $(k = 0, 1, 2, \ldots)$. In z-transform notation, the "transfer function" of the unit is, thus, $1/z$ in analogy to the transfer function $1/s$ of an ordinary integrator in Laplace-transform notation. This fact is directly applicable to the *solution of difference equations* (see also Sec. 6.6).

Figure 6.3.6 illustrates the application of the dual memory unit to the *two-time-scale computation of a double integral.* Similar computer setups are used for the *two-time-scale solution of partial differential equations* with various boundary conditions.[24]

Electronic analog computers with analog storage accessories also permit *multiplexing of computing elements* in the manner discussed in Art. 6.3.4; an interesting example is the DYSTAC study of a four-component distillation column with 30 plates[24] (see also Sec. 5.7). Such a computation would require in excess of 500 integrators with ordinary all-parallel computer operation but was solved within two minutes with 43 operational amplifiers in the DYSTAC system.[25]

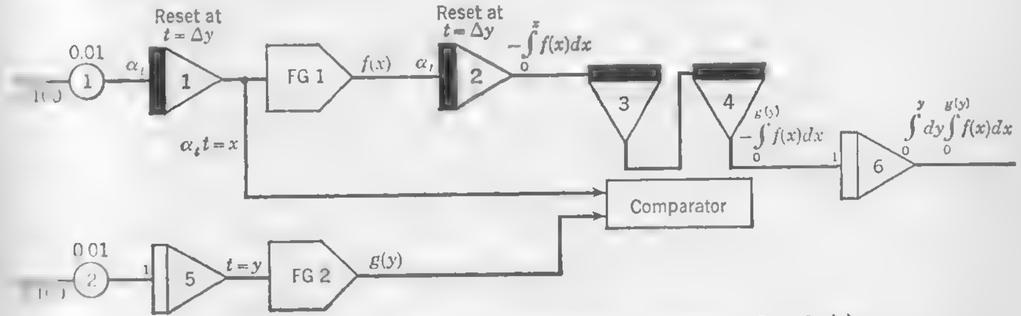


FIG. 6.3.6. Two-time-scale computation of the double integral $\int_0^y \int_0^{g(y)} f(x)\,dx$. The half-black integrator symbols denote "fast" integrators with short time constants $1/\alpha_t$ suitable for repetitive computation. The "fast" integrator 1 generates the "fast" time $\tau = \alpha_t t$ $(\alpha_t = 20)$. The given function $f(x) = f(\alpha_t t)$ appears at the output of function generator 1 and is integrated by the "fast" integrator 2. The input memory integrator 1 tracks $\int_0^x f(x)\,dx$ until $x = \alpha_t t$ equals the given function $g(y)$ of the second variable y, which is represented by the "slow" time $t = y$ (see also Ref. 23).

Finally, the DYSTAC technique permits the synthesis of a *time-delay unit* (see also Art. 6.11) with the aid of a relatively small number of memory pairs whose outputs are combined with an interpolation formula.[24]

Electronic analog computers with electronic storage accessories constitute an important milestone on the way to a true hybrid computer combining the flexible logic and memory of a digital computer with the fast all-parallel arithmetic of the electronic analog computer. Attempts to operate conventional analog computers at high repetitive speeds can cause serious phase-shift errors due to capacitances in cables and phase shift in coefficient potentiometers as well as limitations in the computing elements themselves (Sec. 2.4). This is especially true in the case of solutions involving undamped sinusoidal oscillations. New numerical analysis techniques, yet to be developed, may permit one to check sets of fast repetitive computer runs against more accurate "slow" runs from time to time. It is clear that the new automatic-programming techniques require much more sophisticated programming and numerical analysis than straight analog-computer simulation. In the writer's opinion, the full advantage of the new methods will be attained only after truly integrated systems are designed. Such systems will not only combine high-speed computing elements and new coefficient potentiometers with a minimum of patchbay wiring, but they will also include some digital-computer components, such as logic gates and inexpensive simple and preset decimal-counting units; such digital computing elements can do many jobs much more cheaply than sets of chopper-stabilized analog-computer amplifiers. The

continued development of such techniques may, indeed, lead to something of a breakthrough in the computer art.

References

1. Close, R. N., and G. O. Thogersen: Using an Electronic Analog Memory, *Electronic Design*, December, 1955.
2. Diamantides, N. D.: A Multipurpose Electronic Switch for Analog Computer Simulation and Autocorrelation Applications, *IRE Trans. PGEC*, December, 1956.
3. Lofgren, L.: Predicators in Time-shared Analog Computers, *Proc. International Analogy Computation Meeting*, September, 1955.
4. Blanyer, C. G.: Sampled-data Techniques in Complex Simulation, Eastern Simulation Council Paper, Hydel, Inc., Waltham, Mass., Jan. 5, 1959.
5. Rawdin, E.: Time Multiplexing as Applied to Analog Computation, *IRE Trans. PGEC*, March, 1959.
6. Ragazzini, J. F., and G. F. Franklin: *Sampled-data Control Systems*, McGraw-Hill Book Company, Inc., New York, 1958.
7. Millman, J., and T. H. Puckett: Accurate Linear Bidirectional Diode Gates, *Proc. IRE*, January, 1955.
8. Sebestyen, G.: A Design Technique for Pedestal-free Switching Circuits, *IRE Trans. PGEC*, September, 1957.
9. Gray, H. J., M. Rubinoff, and J. Tompkins: A Diode Multiplexer for Analog Voltages, *IRE Trans. PGEC*, June, 1955.
10. Koerner, H., and G. A. Korn: New Operational-amplifier Techniques for Precision Switching and Function Generation, *Electronics*, Nov. 6, 1959.
11. Bekey, G. A.: Generalized Integration on the Analog Computer, *IRE Trans. PGEC*, June, 1959.
12. Stubbs, G. S.: SADSAC: A Sampled-data Simulator and Computer Using Stepping Relays, *Automatic Control*, April, 1958.
13. Wilkes, S. G.: An Experimental Electrical Analogue for Study of Transient Response of Two-fluid Heat Exchangers, Mechanical Engineering Department, Stanford University, Sept. 1, 1954.
14. Herzog, A. W.: Pulsed Analog Computer for Simulation of Aircraft, *Proc. IRE*, **47**:847 (May, 1959).
15. Lee, R. C.: Logical Design of a High-speed Analog-digital Computer for Simulation, MIT M.S. Thesis, June, 1958.
16. Lee, R. C., and F. B. Cox: A High-speed Analog-digital Computer for Simulation, *IRE Trans. PGEC*, June, 1959.
17. Holmquest, C.: Data Storage, Level Sensing, and Switching in an Analog-digital Function Generator, *MIT Servomechanism Lab. Rept.*, August, 1957.
18. Bourland, H. M.: An Analog-digital Multiplier-divider for Function Generation, MIT M.S. Thesis, August, 1957.
19. Dumas, H. J.: A Pulsed Analog Multiplier-divider, MIT M.S. Thesis, May, 1958.
20. Van der Velde, W. E.: A New High-speed Statistical Computer, GPS Instrument Co., Boston, Mass., 1957.
21. Data Sheet PDA-C7SF (Probability-distribution Analyzer), GPS Instrument Co., Boston, Mass., 1959.
22. Korn, G. A.: Repetitive Analog Computers at the University of Arizona, *Instruments and Control Systems*, September, 1960.
23. Van der Velde, W. E.: Make Statistical Studies on Analog Simulations, *Control Eng.* June, 1960.
24. Andrews, J. M.: Mathematical Applications of the Dynamic Storage Analog Computer, *Instruments and Control Systems*, September, 1960.
25. Application of DYSTAC to Multi-component Destillation, Computer Systems, Inc., Monmouth Junction, N.J., 1960.

PART 4: ELECTRONIC ANALOG COMPUTERS AS CONTROL-SYSTEM COMPONENTS

By W. K. McGregor

INTRODUCTION

6.4.1. Survey. It is reasonable to postulate that, if an electronic analog computer can simulate control system and plant for design purposes, the computer must be able to perform certain actual control functions.

In particular, synthesis of linear control systems leads to mathematical operations which are not usually implemented exactly by conventional controllers. If desired, *integration and other linear operations such as error sensing, addition of signals, differentiation, and phase compensation are readily accomplished with operational d-c amplifiers*, which permit large adjustments in gain, integral time constant, and other parameters simply by substitution of different resistors. Multivariable systems are also handled with relative ease.

Of equal, if not more pronounced, importance are applications of *nonlinear computing elements* to closed-loop systems. These applications fall, roughly, into three categories:

1. Use of function generators to linearize a nonlinear transfer characteristic (e.g., area-vs.-position characteristic of a control valve)
2. Use of multipliers and/or function generators to generate complicated control variables (e.g., combustion efficiency of a jet engine)
3. Nonlinear control action (e.g., variable damping; use of limiters to prevent instability of structural damage)

Many other applications, such as trajectory programming, cross-connection of related systems, and optimalizing control exist (see also Sec. 5.12).

An equally wide area of application involves the so-called *two-time-scale or predictor control systems*. Here the system performance is represented on an electronic analog computer on a fast time scale; solutions are obtained at prescribed intervals from inputs derived from sampled system data, and corrective action is transmitted to the much slower system as changes in controller set points, valve positions, storage levels, or other appropriate input.

ANALOG-COMPUTING ELEMENTS IN DIRECT CONTROL

6.4.2. Linear Operations. The basic block diagram of a feedback control system may be conceived as illustrated in Fig. 6.4.1. If the process, controlling element, and

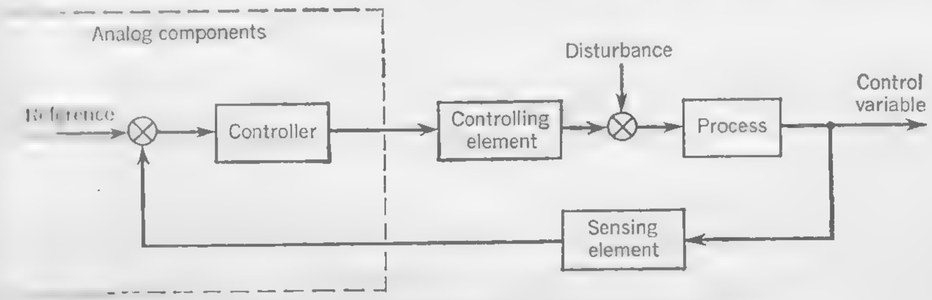


FIG. 6.4.1. Control-system block diagram.

sensing element can be described by linear differential equations, a linear synthesis utilizing any of the many available methods[1] will yield a mathematical specification for the controller block. Conventional modern practice would be to employ commercially available pneumatic, hydraulic, or electronic controllers which best (or most

economically) approximate the mathematical specification. If it is imperative that the linear mathematical function be implemented *exactly* by the controller, the operational computing amplifier offers the most practical solution. An example of such a requirement is the need for zero steady-state error in many systems. This requires type I control action,[2] i.e., the forward loop must contain a pure integration. A d-c analog integrator accurately satisfies this requirement. Table 6.4.1 illustrates the use of analog-computer components for other linear operations.

6.4.3. Nonlinear Operations. Table 6.4.2 lists a number of applications of nonlinear analog-computing elements, frequently replacing complicated special-purpose equipment.

Table 6.4.1. Linear Control Operations

Operation	*Method*
1. Accurate implementation of linear compensation transfer functions	D-C operational amplifiers with appropriate *RC* networks
2. Accurate type I, II, or III control action[2]	D-C operational amplifiers as integrators in the forward loop of the system
3. Large ranges of calibrated controller parameter adjustments	Input or feedback resistor switching or adjustment on operational amplifiers and/or potentiometer setting
4. Lead compensation of feedback element (i.e., thermocouple, pressure transducer, etc.)	Input *RC* parallel network to operational amplifier
5. Monitoring and/or recording of error, proportional action, integral action, controlling element input, etc.	Voltages representing each variable appear as operational amplifier outputs
6. Generation of time functions for input to control systems	Linear or power series in time using chain of integrators
7. Elimination of system interactions in dependent loop processes	Cross-feeding of anticipatory and corrective signals between operational amplifiers

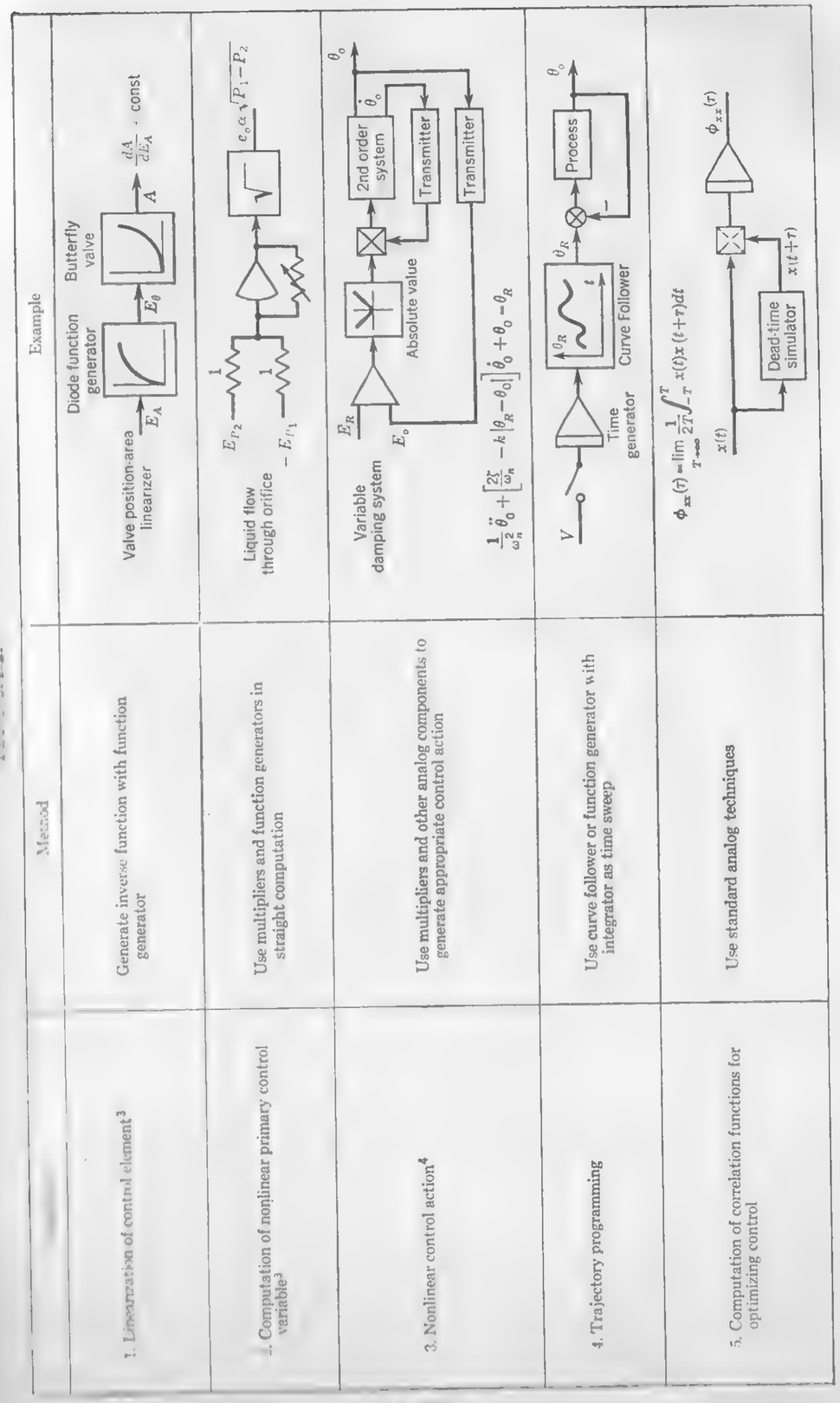
6.4.4. A Specific Application: Gas-flow Process. Figure 6.4.2 illustrates a gas flow process designed to simulate altitude pressures at the exhaust of a jet-engine test cell.[5,6,7] The engine air flow exhausts into a large duct, flows through a set of butterfly throttle-control valves, and is pulled to atmosphere by a compressor system.

The basic variable to be controlled is the simulated altitude pressure at the test-cell exhaust. This pressure must be held constant through large variations of gas flow and compressor performance, and the system must respond very rapidly to set-point changes. Referring to Fig. 6.4.2, the total change in control-system gain to achieve optimum performance is dictated by the large change in K_p and in the nonlinearity of the butterfly-valve control system. Without nonlinear compensation this change can be of the order of 140,000 to 1. The control requirement also dictates that the system possess the characteristics of a type I control system (zero steady-state error).

A secondary control requirement is the prevention of *compressor surge*, an aerodynamic effect similar to airfoil stall which can cause severe mechanical stresses on the compressor components.[8] To prevent surge, the compressor operating point must be kept to the right of the vertical line shown on the compressor performance curve in Fig. 6.4.2; when engine air-weight flow is less than the surge value, the control system must rapidly provide by-pass flow. The surge-prevention system is also shown in Fig. 6.4.2.

It is frequently difficult to predict the behavior of a large-scale process accurately during design. Again, for plants already built, transfer-function measurements may require much costly nonproductive plant time. Hence the control system should be sufficiently flexible to permit changes after installation; this is true for the system of Fig. 6.4.2. In addition, this system illustrates several features possible with analog controllers:

1. *Accurate proportional-plus-integral control action* to assure zero steady-state error
2. *Sensing of the logarithm of pressure* to compensate for process-gain variations with operating point



	Method
1. Linearization of control element[3]	Generate inverse function with function generator
2. Computation of nonlinear primary control variable[3]	Use multipliers and function generators in straight computation
3. Nonlinear control action[4]	Use multipliers and other analog components to generate appropriate control action
4. Trajectory programming	Use curve follower or function generator with integrator as time sweep
5. Computation of correlation functions for optimizing control	Use standard analog techniques

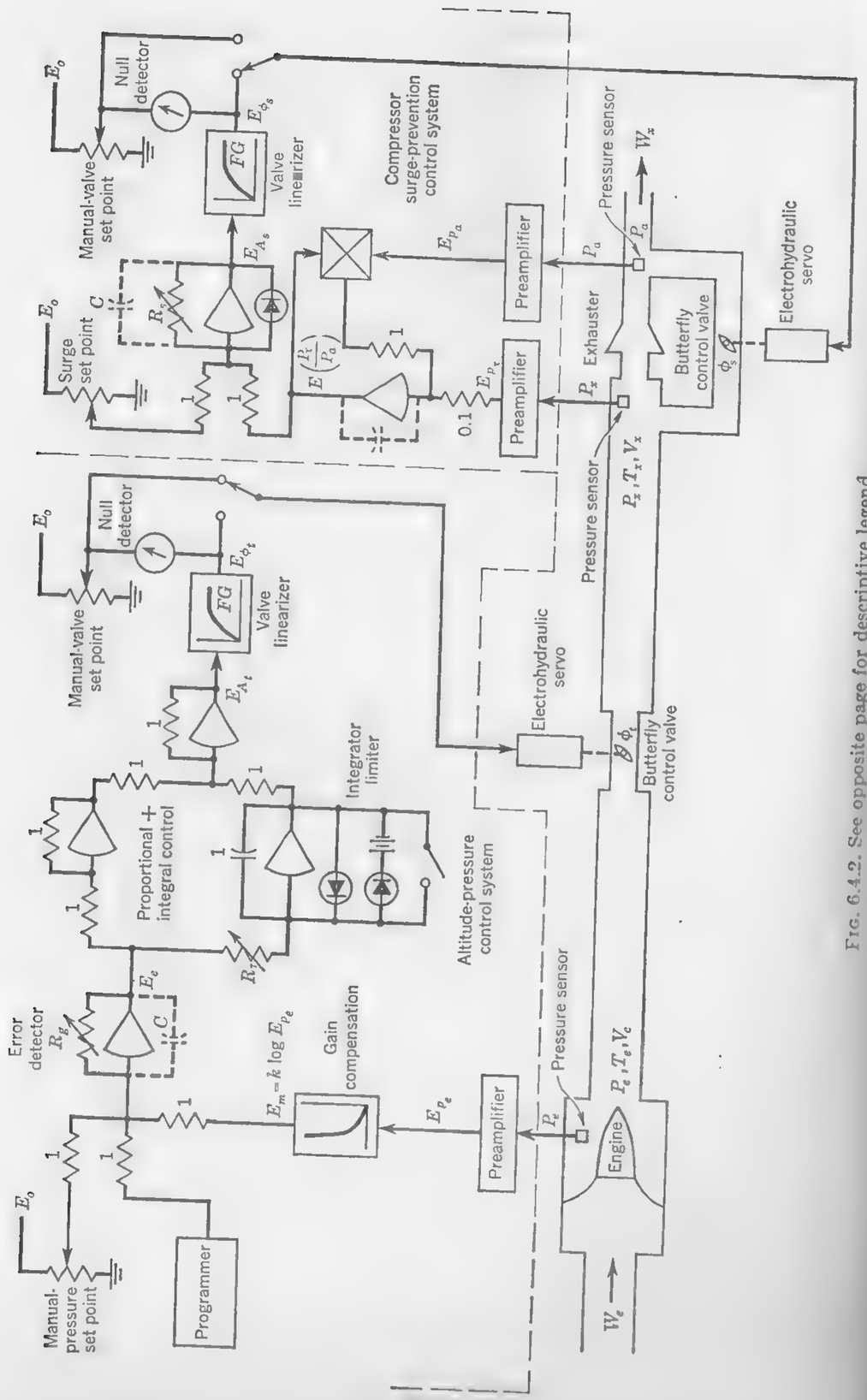


FIG. 6.4.2. See opposite page for descriptive legend.

3. *Linearization of the control-valve to position-area relationship* by use of a function fitter
4. Sensing of pressure *ratio* for compressor surge prevention
5. *Trajectory programming* of altitude, utilizing a curve follower
6. *Monitoring of internal control variables* with a d-c voltmeter
7. *Accurate methods of setting* proportional gain and integral time constant

6.4.5. Computer Requirements. (*a*) *Amplifier Open-loop Gain.* Amplifier open-loop gains as low as 1,000 can often be tolerated in control applications, as compared with the much larger gains required in multipurpose computers.

(*b*) *Apportionment of System Gain.* Amplifier voltage ratings must not be exceeded by the control action. Thus, the major portion of system gain must be in those components where signal magnitudes are small; integrators which might be saturated by error voltages can be equipped with diode limiters (Figs. 6.4.2 and 6.4.3*a*).

(*c*) *Arrangement of Mathematical Operations.* In most applications requiring computer control it is desirable to have independent adjustment of proportional gain, integral and derivative time constants (Fig. 6.4.3*a*). If only occasional adjustments are required, a simpler system like that in Fig. 6.4.3*b* may suffice.

(*d*) *Noise.* In analog controllers controlling relatively slow processes, the effects of a-c noise pickup are easily minimized by small feedback capacitors around operational amplifiers (dash lines in Fig. 6.4.2). Such capacitors may also improve high-frequency stability.

(*e*) *Drift.* D-C amplifier drift must be minimized even inside control loops if nonlinear operations are required. Chopper stabilization will minimize drift and also obviate the need for frequent amplifier balancing. Power supplies with 0.1 per cent regulation and 3 mv ripple have been found satisfactory.

(*f*) *Construction and Packaging.* Conservative component ratings, good ventilation, and, increasingly, transistorization of electronic components (Sec. 7.1) make for reliable operation. Plug-in modules simplify maintenance. Great accuracy is frequently unnecessary. Small removable patchboards assembled with banana plugs and jacks or clip-in terminals permit flexibility of control functions with properly equalized amplifiers. Some facilities use small portable multipurpose analog computers as temporary control-system components.

COMPUTERS IN INDIRECT CONTROL

6.4.6. Indirect Control. In indirect-control applications[9,10,11] the analog computer gathers data from the process and uses past experience to make decisions regarding

FIG. 6.4.2. A jet-engine test facility controlled by an analog process-control system. The control system employs electrohydraulic servomechanisms with frequency response flat to 12 cycles per sec; the actuating d-c signals vary between 0 and 20 volts. The exhaust pressure (controlled variable) is related to the valve displacement ϕ_t by a highly nonlinear differential equation, approximately

$$\tau(P_e, T_e, W_e)\frac{dP_e}{dt} + P_e = K(P_e, T_e)\phi_t$$

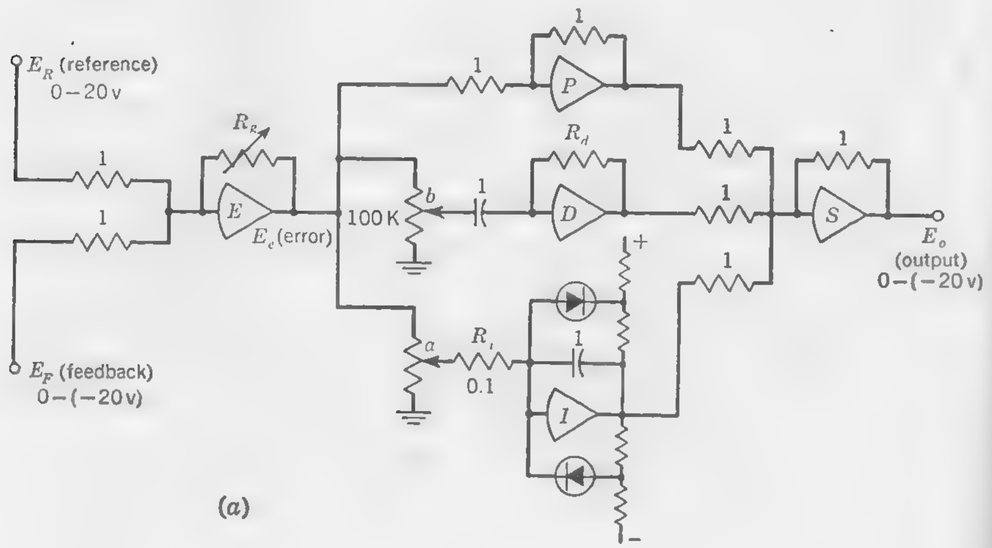
where τ varies between 1 and 60 sec and K varies from 0.001 to 140 lb per sq ft/deg valve rotation. The gain-compensating and valve-linearizing function generators shown compensate for the variation in τ and K and effectively permit the controller to work with the linear "compensated" process transfer function,

$$\frac{E_m(s)}{E_{A_t}(s)} = \frac{K'}{\tau' s + 1}$$

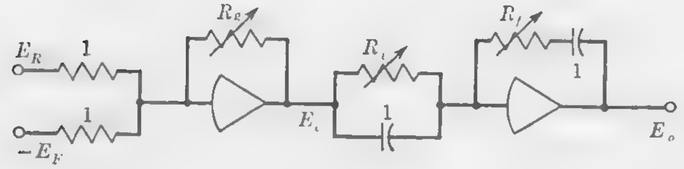
where K' and τ' are constants.

The simple two-mode controller has the transfer function $R_0(1 + 1/R\tau s)$. The following symbols are used with appropriate subscripts:

P	pressure	ϕ	control-valve travel
T	(absolute) temperature	E	voltage signal
W	gas-weight flow	R	resistance



(*a*)



(*b*)

FIG. 6.4.3*a*, *b*. Three-mode analog controllers with noninteracting parameter settings (*a*), and with interacting parameter settings (*b*). The controller transfer functions are, respectively,

$$R_g\left(1 + bR_d s + \frac{a}{R_i s}\right) \quad \text{and} \quad R_g R_f\left(\frac{R_i + R_f}{R_i R_f} + s + \frac{1}{R_i R_f s}\right)$$

controller parameters or logic. The computer thus acts in the manner of a general-purpose analog computer which communicates with the process through suitable transducers (see also Sec. 5.12). Figure 6.4.4 shows an application of two-time-scale control to a flood-control problem.

CONCLUDING REMARKS

6.4.7. Economic Considerations. In medium-to-large plants exemplified by the engine-test facility illustrated in Fig. 6.4.2, the computer controller constitutes only about 0.02 per cent of total cost; this percentage increases as facility costs decrease. In some applications analog controllers are more economical than even pneumatic systems.

On the other hand, very complex two-time-scale control systems for large plants may be handled more economically by digital computers.

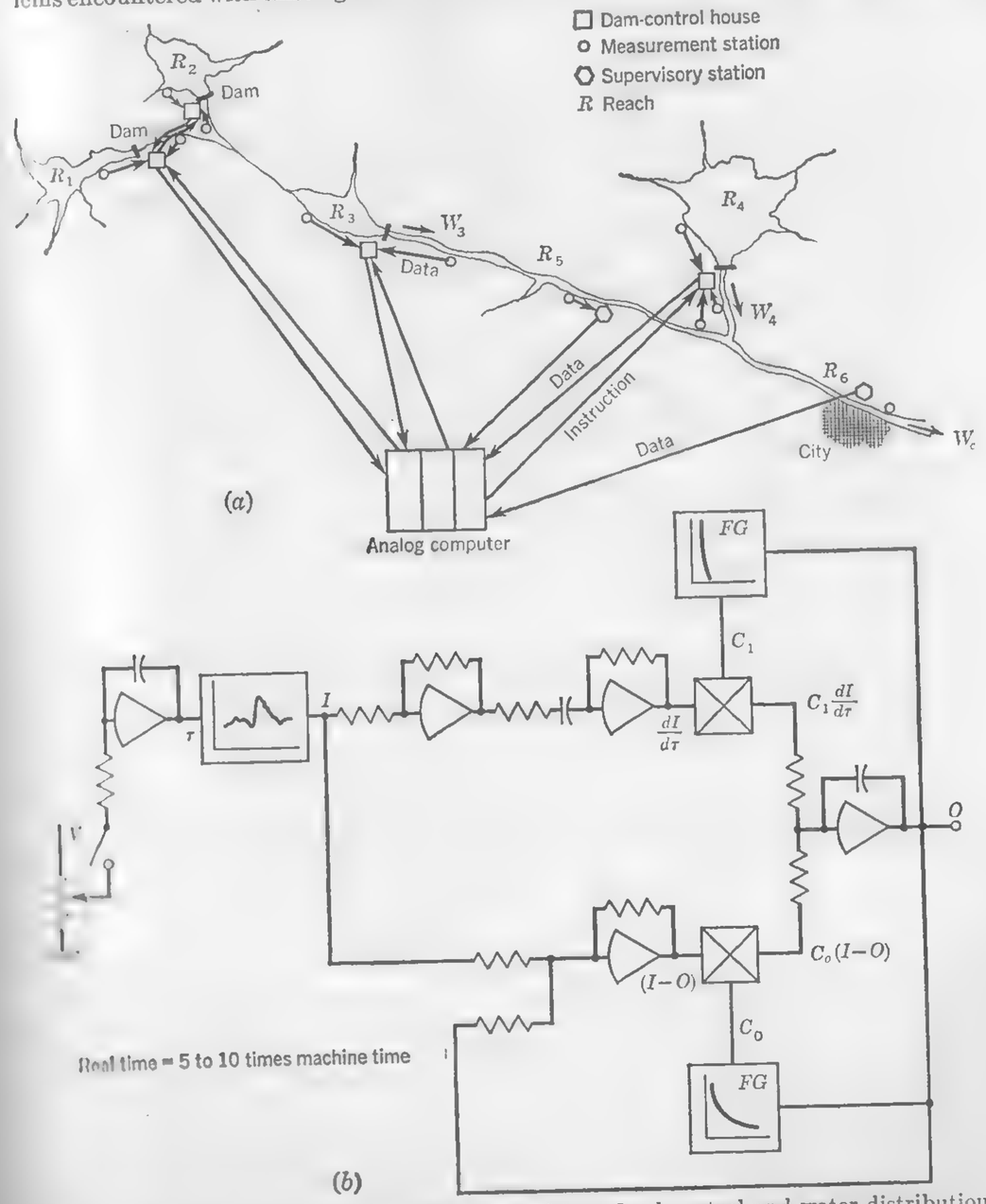
6.4.8. By-products of Analog Control. The example analog-control system of Fig. 6.4.2 also illustrates some of the more or less intangible by-products of analog control, viz.,

Operator training
Rapid isolation of malfunctioning components
Replacement of poorly performing components without system redesign

Extreme flexibility for system modification or for cross connection of two or more interacting systems

Availability of convenient variables for measurement of system performance

6.4.9. Suggested Developments. Analog-control methods can solve various problems encountered with existing control methods. For example, multiple control-signal



(*a*)

(*b*)

FIG. 6.4.4. Application of two-time-scale control to a flood-control and water-distribution problem (Ref. 9). The inflow I and outflow O for a "reach" are related by the differential equation

$$\frac{dO}{dt} = C_0(I - O) - C_1\frac{dI}{dt}$$

where C_0 and C_1 are empirically determined functions of O. Solutions for different hypothetical flow conditions yield an acceptable set of control instructions.

channels are easily implemented and could greatly increase control reliability for critical processes. Another well-suited application is in the switching of control parameters and even methods of control as dictated by the state of the process. "Adaptive" and "optimalizing" control are also within the capabilities of both analog and digital computers. There is a very real requirement for rugged and simple computing elements and matched transducers specifically designed for computer control.

References

1. Truxal, John G.: *Control System Synthesis*, McGraw-Hill, New York, 1955.
2. Chestnut, Harold, and Robert W. Mayer: *Servomechanisms and Regulating System Design*, vol. 1, John Wiley & Sons, Inc., New York, 1951.
3. McGregor, W. K.: Linearization Functions in Process Control Systems, ISA Paper 56-33-3, presented in New York, September, 1956.
4. Lewis, John B.: The Use of Nonlinear Feedback to Improve the Transient Response of a Servomechanism, AIEE Paper 52-199, May, 1952.
5. McGregor, W. K., D. W. Russell, L. F. Burns, and R. Messick: Analysis of Gas Flow System for Dynamic Control Purposes, AEDC-TR-55-11, ASTIA Document AD 88130, Tullahoma, Tenn., April, 1956.
6. Schwent, G. V., W. K. McGregor, D. W. Russell, and L. F. Burns: A New Approach to Control System Design, *ISA Journal*, July, August, September, October, November, and December, 1956.
7. McGregor, W. K., and L. F. Burns: The Dynamic Performance of ETF-B Climatic Control Systems, AEDC-TR-58-11, ASTIA Document AD-204622, Tullahoma, Tenn., September, 1958.
8. Russell, D. W., W. K. McGregor, and Lee F. Burns: The Analog Computer as a Process Controller, *Control Eng.*, September, 1957.
9. GEDA Application to Flood Control and Water Distribution Problems, Goodyear Aircraft Corporation, GER-6053, May, 1954.
10. Zeibolz, H., and H. M. Paynter: Possibilities of a Two-time Scale Computing System for Control and Simulation of Dynamic Systems, *Proc. Natl. Electronics Conf.*, **9** (February, 1954).
11. Eckman, D. P., and I. Lefkowitz: Optimizing Control of a Chemical Process, *Control Eng.*, September, 1957.
12. Hart, C. E., L. M. Wenzel, and R. T. Craig: Experimental Investigation of Temperature Feedback Control Systems Applicable to Turbojet Engine Control, *NACA Tech. Note* 3936, Cleveland, Ohio, March, 1957.

PART 5: COMBINED USE OF ANALOG AND DIGITAL COMPUTERS

By R. M. Leger *and* Jerome L. Greenstein

INTRODUCTION

6.5.1. Analog and Digital Computation. The all-parallel "arithmetic" of an electronic analog computer yields *high-speed solution of linear and nonlinear differential equations with moderate accuracy; real-time simulation* and *partial system tests* are simple. Digital computers excel in *accurate large-scale algebraic manipulations* and permit *convenient generation of multivariate functions* as well as *programming of complex subroutines, logical decisions,* and *iterative solutions.* The two types of computers can often complement each other. Note, incidentally, that many analog computers have digital read-out and programming features (Secs. 4.2 and 6.3).

Many important problems can benefit from the *combined but separate* use of analog and digital computation. *Digital solution checks* are an obvious example. Again, an analog-designed missile autopilot system may be substituted as a transfer function in the digital-computer study of a guidance system requiring accurate trajectory computations.

6.5.2. Combined Analog-digital Simulation. If, in the last example, the natural frequencies of autopilot and guidance system are not sufficiently separated or decoupled, these systems may interact radically. In this case accurate guidance-system design will require simulation by *combined* analog and digital computers. Frequently real-time simulation and partial system tests are still possible.

Conversion devices are required to transfer information between analog and digital computers. These devices must perform the functions of *analog/digital/analog conversion, sampling and holding of analog data,* and *buffering of digital data* (Arts. 6.5.5 to 6.5.8).

NUMERICAL TECHNIQUES USED IN DIGITAL SIMULATION

6.5.3. Sampled-data Interpolation. A digital computer inserted into an analog-computer setup must necessarily work with *sampled data.* Thus, in the simulation of continuous phenomena the process of interpolation is fundamental to the solution, particularly if integrations must be performed.[1–5] Simple linear interpolation is usually not sufficiently accurate unless very high sampling rates are used. Quadratic or cubic curve fits usually provide a considerable improvement, but they require the use of more data at each step. This additional data may be in the form of additional past data points or various functional derivatives of the data. A Taylor-series extrapolation is an example of the use of many derivatives at a single point.

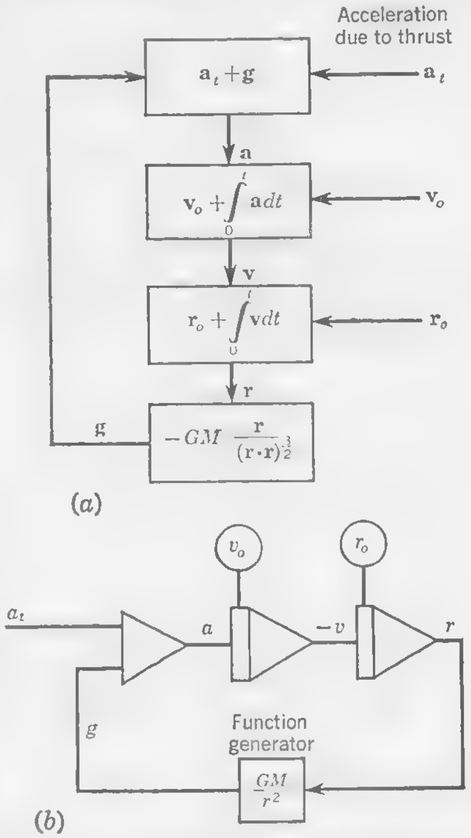


FIG. 6.5.1. Signal flow for solution of missile trajectory equations (round earth, no atmosphere): three-dimensional digital solution (*a*), and one-dimensional analog solution (*b*).

6.5.4. Digital Solution of Differential Equations. A digital computer solves differential equations by using step-by-step *numerical integration* of the sampled data. The integration formulas are based upon the interpolation processes discussed in Art. 6.5.3. Those which make use of past data points require special handling at the start of a problem, when no past data exist. Since the digital computer carries out operations in sequence instead of in parallel, special techniques are also required to account for the instantaneous dependence of the integrands on the outputs of the integrators (Fig. 6.5.1). This is the digital equivalent of the feedback loops on a differential analyzer. It can be handled by iterative methods, such as the predictor-corrector technique. With this technique the step size can be automatically adjusted to control the error in the process. Another widely used technique is the Runge-Kutta method which handles the "feedback loop" and the integration in a unified process which uses no past data and thereby avoids the start procedure problem. For the representation of linear filters with specified transfer functions it is possible to develop special recursion formulas to yield output directly in terms of inputs and past outputs without iteration.[9]

In all numerical calculations, one must guard against accumulation of errors. *Errors due to finite sampling rates* act somewhat like bandwidth limitations in analog-computing elements and may introduce instability.[4,8]

Special precautions are necessary if the simulation requires *differentiations* and handling of *discontinuous functions.*

CONVERSION EQUIPMENT

6.5.5. Digital-to-analog Converters (Decoders). Each data word from a digital computer is normally sent in the form of binary-coded signals on a parallel set of wires originating from a flip-flop register. For example, assuming a data word consists of n binary digits, then on each one of the n wires there will be an electrical signal representing either a binary "one" or a binary "zero." The decoder must convert this data word into an analog voltage (Fig. 6.5.2).

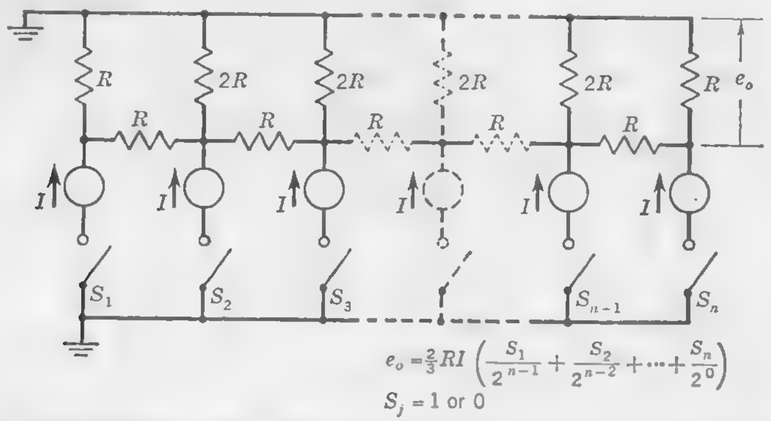


FIG. 6.5.2*a*. Voltage decoder with ladder network. Each current source generates a current I of equal magnitude into a load of $\frac{2}{3}R$. Each switch, S_1 to S_n, is closed by a binary "one" signal appearing on the corresponding output wire of the digital computer.

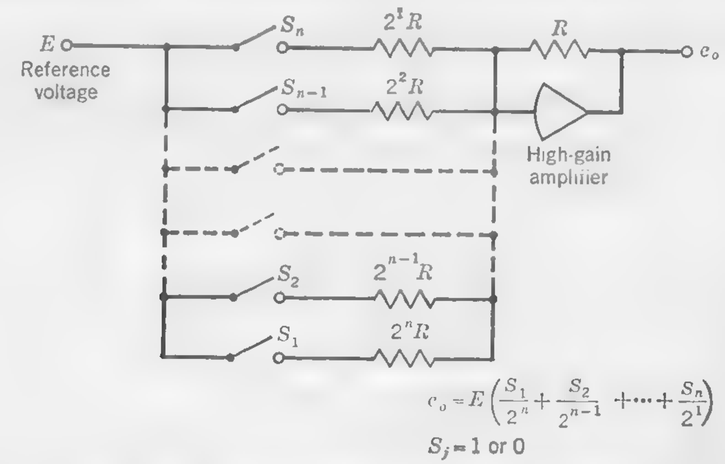


FIG. 6.5.2*b*. Voltage decoder with weighted resistors.

6.5.6. Analog-to-digital Converters (Encoders). Figures 6.5.3 and 6.5.4 show a typical *shaft-position encoder* (positioned by a computer servomechanism) and two *voltage encoders.*

6.5.7. Analog Sample-hold Circuits. *Sample-hold circuits*, described in Sec. 6.3, sample an analog voltage and hold it until the conversion has been completed. If the conversion device is sufficiently fast compared with the rate of change of the analog variable, then the sample-hold circuit may not be required. Sample-hold circuits also permit the *time-sharing* (*multiplexing*) of one analog-to-digital converter for several analog variables.

6.5.8. Buffering of Digital Data. A digital buffer-storage register is required when different pieces of digital equipment not operating synchronously are connected together. For example, after analog-to-digital conversion, the digital data must be

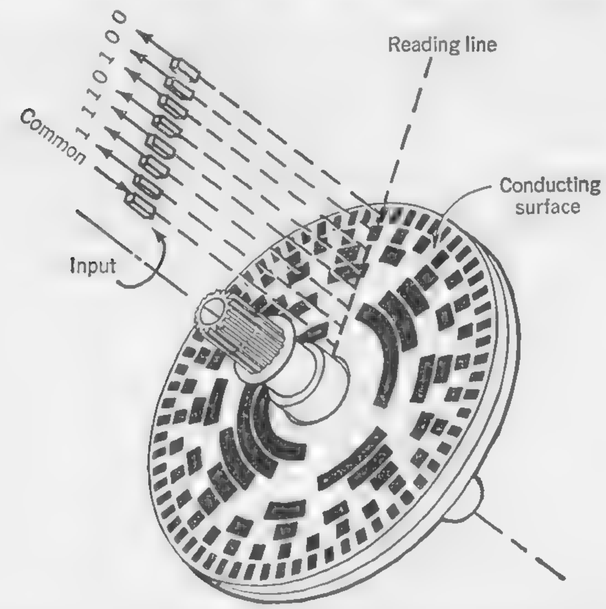


FIG. 6.5.3. Brush-type or photoelectric shaft-position encoders generate a unique code word on parallel outputs for each shaft position; here $2^7 = 128$ binary numbers represent a full revolution. (*From R. M. Leger and J. L. Greenstein, Control Engineering, September, 1956.*)

temporarily stored until the digital computer is ready to accept the data. Storage is normally accomplished by a flip-flop or magnetic-core register.

Buffering circuitry also serves the purpose of converting the digital output of the encoder into electric signals compatible with the digital-computer input-output circuitry.

6.5.9. Synchronization. A source of accurately timed *synchronizing pulses* is required to initiate sampling of analog data and the desampling of digital data. The starting of the analog computer must also be synchronized with sampling and desampling (see Art. 6.5.17 for an example).

The synchronizing pulses can also be used to initiate the digital-computer input-output routine required to transfer data between the digital computer and the conversion equipment. Since a digital computer can normally handle only one binary word at a time, special storage registers external to the computer will be required to hold the data words until the computer is ready to accept them. The actual synchronizing and buffer circuitry required will depend upon the particular computer used.

A special *interrupt* routine may enable the digital computer, upon receipt of an external "interrupt" pulse, to discontinue its current sequence of instructions and jump to a new sequence (e.g., a set of input-output commands). The computer can thus perform other computations while waiting for sampled analog data.

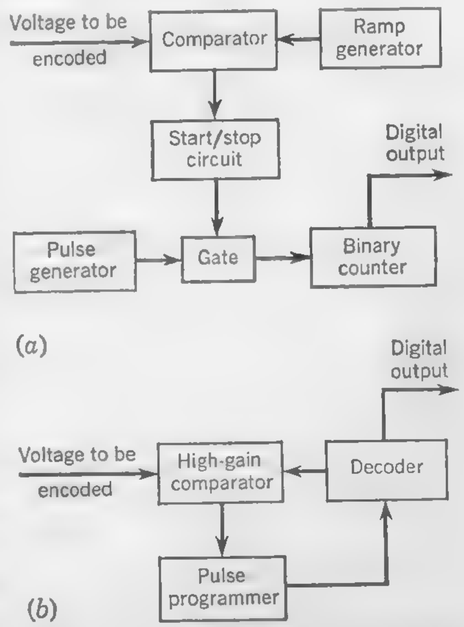


FIG. 6.5.4. Voltage encoders: voltage-to-time-to-digital encoder (*a*), and feedback encoder (*b*).

6.5.10. Conversion Errors. (*a*) *Static Error.* The static accuracy of an analog-to-digital converter is, of course, no better than that of the analog signal fed into it. For this reason, most converter systems have been limited to 10- to 12-bit word lengths corresponding to accuracies of approximately 1 part in 1,000 to 5,000 of full scale. In some cases, the converters may be required to handle signals (e.g., guidance errors), which are large during part of the problem but are driven to zero at the end, and the over-all merit of the system is judged by the accuracy with which the final zero is achieved. If such problems are to be handled without range switching (see also Sec. 5.5), a longer word length is useful to prevent overly coarse quantization errors when operating close to zero. Converter systems have been built with 18-bit words to provide a dynamic range of 100,000 to 1 for such purposes. It is, of course, necessary that the analog computer have sufficiently low noise characteristics if full advantage is to be taken of a system with such wide dynamic range.

(*b*) *Dynamic Errors.* Dynamic errors are introduced by timing inaccuracies and smoothing filters. Consider the problem of converting into digital form the value at time t of an analog voltage which is changing at a rate of K volts per sec. In principle, it is necessary to sample the analog voltage at time t and then convert the sampled value into digital form. If the voltage is actually sampled at $t + \delta t$, the resulting value will be in error by the amount $K\ \delta t$. A somewhat similar effect takes place with respect to the timing of the presentation of converted digital data to the analog computer at the digital-to-analog converters.

A dynamic error similar to the above may also be introduced if the bandwidths of the analog portions of the converters are unduly limited by ripple-smoothing networks. The permissible error due to this effect may be expressed in terms of the error which would be produced by a low-pass filter with a specified time constant. The error due to such a time constant varies with rate of change of the signal in essentially the same manner as for a timing error.

PROBLEMS PECULIAR TO COMBINED ANALOG-DIGITAL SIMULATION

6.5.11. Sampling Problems. The selection of a minimum sampling frequency is normally based upon the bandwidth of the analog data. Consideration must be given to the effect of sampling on system stability; the effect of low sampling rates is somewhat like that of bandwidth-limited analog-computing elements. Hence a given digital computer may not be able to handle rapidly varying variables in real-time simulation.

Before sampling, special care must be taken to filter out extraneous high-frequency noise on the sampled quantity. Although in most analog simulations the presence of high-frequency noise might be hardly noticed, the sampling process will convert the noise into lower frequencies which will lie in the same spectral range as the desired data.

6.5.12. Desampling. Data flow from the digital to the analog computer necessarily involves the process of *desampling.* This is basically a process of interpolation which supplies data to the analog computer at all points between the given digital samples on the basis of some reasonable assumptions as to the character of the quantity being simulated.

As a starting point, some device must be provided for maintaining an output voltage between the times when an output is being obtained from the digital computer. Furthermore, provision must be made for storing the data between the time they are delivered by the digital computer and the time they are needed by the analog computer. Consider the simple rectangular or step interpolation shown in Fig. 6.5.5. The value of x presented to the analog computer must be up-dated at each of the times $t_1, t_2, t_3, \ldots, t_j$. The digital computer must generate the up-dated values corresponding to each t_j at some arbitrary time prior to the particular t_j involved (small circles in Fig. 6.5.5). The irregularity of the times at which the various digital values are produced has been exaggerated in Fig. 6.5.5 for emphasis.

It is apparent from Fig. 6.5.5 that provisions must be made for holding both the old and new data during the period between the time when the digital computer delivers new data and the time when the analog computer is ready to make use of it. In certain types of problems it may be possible to synchronize the digital computer adequately so as to cause the read-out time to come sufficiently close to t_j to eliminate the need for new data storage outside the computer, but this is not always possible and places an additional constraint on the digital program. If the digital-to-analog converter is of the type shown in Fig. 6.5.2*a*, driven by a flip-flop register or equivalent, the holding of the analog-computer input step can be accomplished directly by the converter. The storage of new data prior to transferral to the analog computer requires some form of buffer storage. An alternate method would be to store the new data in the ladder-network register and to use analog sample-hold units to provide the final analog-computer input. The use of a sample-hold device is, in principle, less accurate than the digital storage, but it may be sufficient and more convenient in some cases.

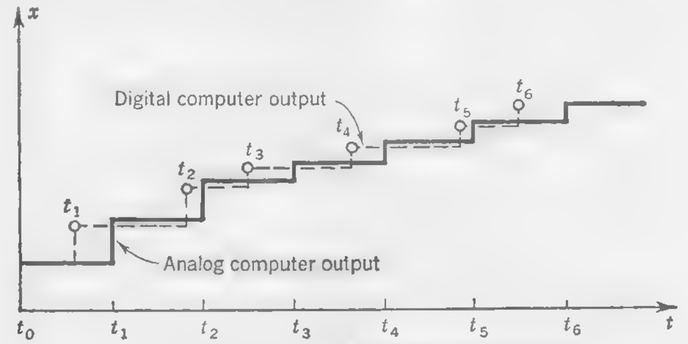


FIG. 6.5.5. Step-type desampler behavior.

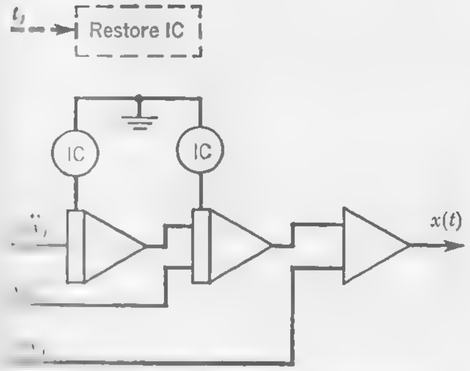


FIG. 6.5.6. Taylor-series interpolation.

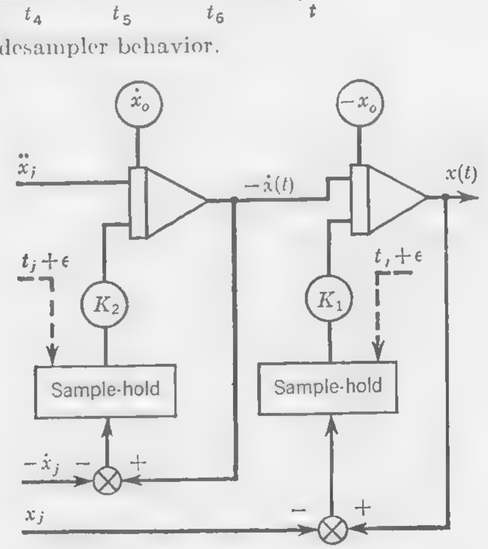


FIG. 6.5.7. Taylor-series desampler with continuous output.

Once the basic facilities for generating the step input of Fig. 6.5.5 have been provided, more refined interpolations can be instrumented by interconnections of additional analog equipment and, perhaps, by use of additional digital-to-analog conversion channels. As a simple example, the Taylor-series interpolation scheme of Fig. 6.5.6 requires the use of one conversion channel for each term desired in the Taylor series. It also requires that the integrators be modified to permit driving their initial conditions back to zero by means of a pulse from the master timer at each t_j. These integrators need not have low-drift characteristics since they are used only over a short time span.

Figure 6.5.7 shows an alternative method of instrumenting the Taylor-series interpolation. This method eliminates the discontinuous correction of $x(t)$ at each t_j, since

all corrections are introduced through integrators. It does not require the modified integrators but makes use of sample-hold devices instead.

In principle, multiple-point interpolation could be carried out with a single digital-to-analog channel through storage of past data points in the analog computer. Direct application of this process becomes very cumbersome and of questionable accuracy. However, a nearly equivalent effect can be achieved by use of an integral-control servo system (Fig. 6.5.8). Comparison of Figs. 6.5.7 and 6.5.8 shows the latter to be similar to a two-term Taylor-series interpolation, where $\dot{x}$ is obtained from past data by the extra integrator.

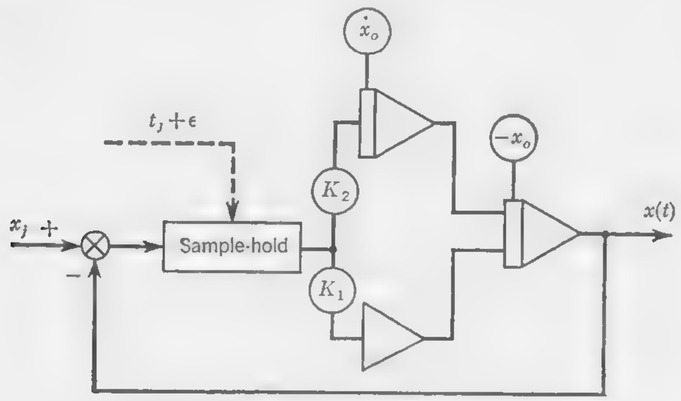


FIG. 6.5.8. Integral-control servo desampler.

6.5.13. Digital-computer Speed and Time Delay. Since the digital computer must deliver new output data in every sampling interval, it must be capable of carrying out all the necessary computations within the sampling period. If the digital computations must be stopped in order to read data into or out of the computer, then within each sampling interval τ there must be sufficient time to allow:

1. Reading input data
2. Carrying out a complete computation step
3. Delivering output data

If the time interval τ is not sufficient to permit the above steps to be carried out, several obvious remedies may be applicable, e.g.,

1. Increase τ
2. Slow down the analog-computer time scale
3. Decrease the amount of digital computation
4. Use faster digital-computing equipment

A more subtle approach is to split the digital computation into "high-frequency" and "low-frequency" aspects, and to compute only the high-frequency effects completely in each time interval τ. The low-frequency computations can be made on a larger interval τ' so that only a fraction of these need be carried out in the basic interval τ. Simple interpolations of the low-frequency variables can be used where needed as parameter inputs to the high-frequency computations.

It should be noted that the time required to make a data conversion from analog to digital or vice versa is *not* part of the speed problem just discussed. The only pertinent considerations in this respect are the digital read-in and read-out times, and this only if the computer must interrupt computations during read-in and read-out. On the other hand, the conversion times, as well as the read-in and read-out times, are unavoidably a part of a related problem—that of an erroneous *time delay* seen by the analog computer.

In a typical combined analog-digital simulation of a guidance system (Art. 6.5.15), the digital computer forms part of the feedback loop around the analog portion of the problem. At some time t_a the analog data are sampled, converted to digital, and read into the digital computer. At some later time the digital computer is ready to deliver its output data back to the converters which carry out the conversion and present the

appropriate voltages to the analog computer at a time t_b. The digital computer has produced output data valid for time t_a, but it is not available to the analog computer until some later time t_b. This is equivalent to inserting a time delay $(t_b - t_a)$ into the feedback loop. Since time delays can have a serious effect upon the response of a servo system, this delay must be handled carefully if the results are to be meaningful. One of the simplest ways to eliminate or reduce this type of delay is to perform an extrapolation in the digital computer, so that the output data are valid for time t_b. For small corrections this is often a very satisfactory technique. On the other hand, extrapolations tend to become less and less accurate as they are extended farther ahead of the data on which they are based. Also, extrapolations are valid only over such time spans as are small with respect to the period of the highest frequency existing in the data. For example, an extrapolation will fail to show any response to inputs occurring between t_a and t_b; if quick response to new inputs must be represented, there is no substitute for low delay in the system. As before, it may be possible in some problems to split the digital computations into low-frequency and high-frequency phases, where the short delay time is critical only in the high-frequency phase. Alternatively, it may be possible to keep the rapid-response high-frequency part of the computation entirely within the analog computer.

6.5.14. Simulation Setup. The major portion of setup time prior to the running of a combined analog-digital problem involves the analog computer. Since the actual amount of analog setup time is not always predictable, the digital computer must be kept busy on other problems until it is needed. It is desirable to have the digital computer available at a moment's notice. This can be done only by using it for short problems prior to the time it is expected to be needed, or by using an "interrupt" feature. If analog-computer or conversion-equipment trouble develops during the simulation, the "interrupt" feature enables the digital computer to return quickly to the interrupted problem.

Because of the expense of tying up both analog and digital facilities, it is essential that the simulation program be arranged so that a maximum amount of check-out and debugging be accomplished independently on each computer. To facilitate this, the program on each computer should provide for a simplified simulation of the other computer. In particular, check-out of the conversion link can be greatly facilitated by the use of a "digital-computer simulator" to simulate the input-output register of the digital computer.

It is obvious that combined analog-digital simulation involves considerably more computer equipment than either analog or digital simulation alone. Over-all operating reliability is somewhat decreased when compared with purely digital simulation. When compared with analog simulation operating reliability can often be increased, since some of the more difficult analog operations can be transferred to the digital computer.

EXAMPLE OF A COMBINED ANALOG-DIGITAL SIMULATION

6.5.15. A Missile-system Problem. The missile-system problem of Fig. 6.5.9*a* illustrates two important applications of combined analog-digital simulation, viz., (1) simulation of a sampled-data control system which intrinsically involves digital computation, and (2) use of the digital computer as an accurate and flexible multivariable function generator to aid in analog computation.

The missile system of Fig. 6.5.9*a* comprises (1) a ballistic missile, which is to be guided into a desired ballistic trajectory, (2) a ground-based radio tracking system and communication link which measures missile velocity and position and transmits steering and engine-cutoff commands, (3) a digital-guidance computer, and (4) an autopilot which keeps the missile stabilized in flight by controlling the deflection of the missile's rocket motors. During the first few minutes of flight, the rocket motors are powered, and the guidance computer must (1) sample the data from the tracking system, (2) filter out tracking-system noise, (3) compare the actual trajectory with the required trajectory, (4) generate pitch and yaw steering commands for course correction, and (5) cut the rocket motors when a desired ballistic-trajectory initial veloc-

ity has been reached. Angular-velocity steering commands are transmitted to the missile autopilot n times per second via the communication link, and tracking data are sampled by the computer at the same rate.

6.5.16. Combined Simulation. Figure 6.5.9*b* illustrates the combined analog-digital simulation of this system, under the assumption that the guided portion of the flight under study takes place at an altitude so high that aerodynamic forces are negli-

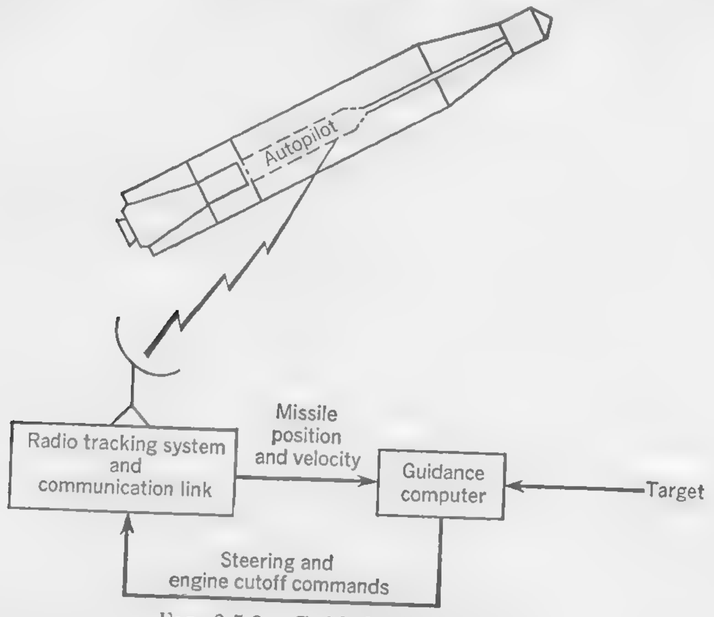


FIG. 6.5.9*a*. Guided-missile system.

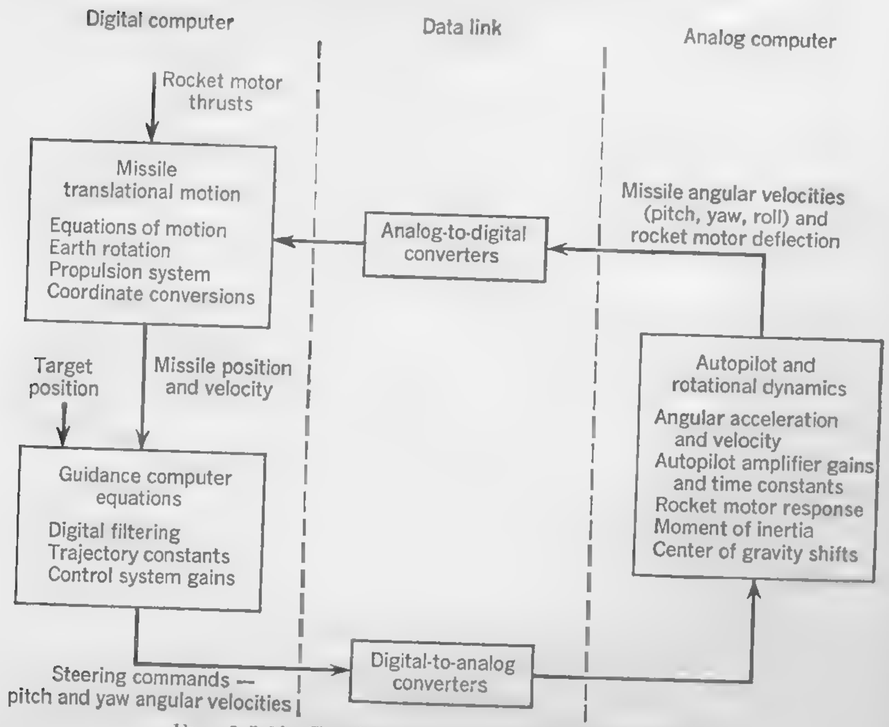


FIG. 6.5.9*b*. Combined analog-digital simulation.

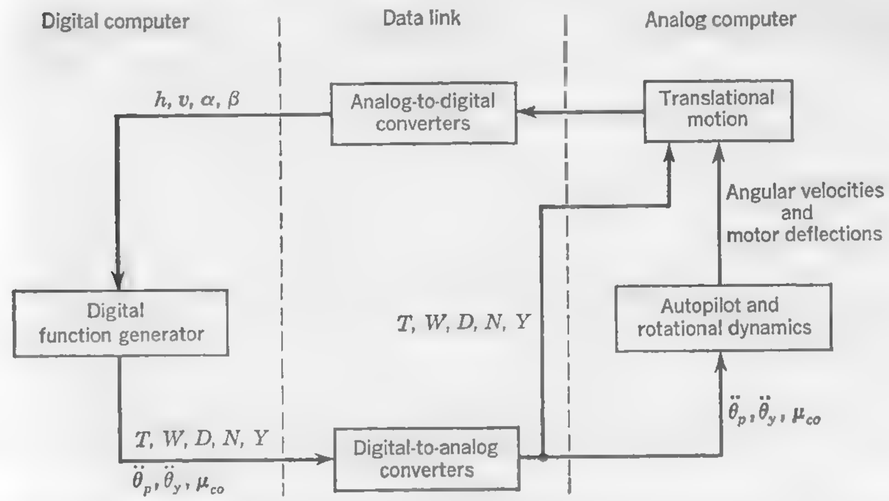


FIG. 6.5.9*c*. Analog missile simulation with the digital computer used as a function generator. Interpretation of symbols:

T, W, D, N, Y	Thrust, weight, aerodynamic forces
$\ddot{\theta}_p, \ddot{\theta}_y, \mu_{co}$	angular accelerations, control moment coefficient
h, v, α, β	altitude, velocity, angles of attack

gible. Note that *the missile trajectory is computed on the digital computer.* Even though the translational motion of the missile is naturally continuous, the accuracy of an analog computer is inadequate for precise determination of the missile's trajectory and impact point.

The position and velocity information fed to the simulated guidance equations represents the sampled data from the tracking system. The pitch and yaw angular-velocity steering commands from the guidance equations are converted into voltages and then fed to the analog-simulated autopilot. These steering commands, computed n times per second, are fed to the analog computer as a staircase function, each new step occurring each $1/n$ sec to simulate the transmission of the commands in the actual system. *The simulation of the autopilot and the rotational dynamics, because of their lower accuracy requirements and their higher relative frequency components, are well suited for handling on the analog computer.*

Closing the loop, the missile angular velocities (pitch, yaw, roll) and the rocket-motor deflection angles are sampled by the data link and fed to the digital computer. These sampled functions are used to resolve the thrust of the rocket motors into rectangular components with respect to an inertial-coordinate system.

6.5.17. Sequence of Computer Operations. The sequence of computer operations for the simulation of this missile system might be as follows:

1. Analog computer equipment is set up.
2. When time for this problem becomes available on the digital computer, the computers are physically interconnected.
3. The digital computer performs its initial sequence of instructions.
4. At the completion of 3, a "start" pulse from the digital computer starts a train of synchronizing pulses with a repetition rate of n cps. The first synchronizing pulse alerts the analog computer.
5. After $1/n$ sec of analog and digital computation, the second synchronizing pulse arrives, initiating the digital computer input-output routines, and starting the analog-to-digital conversions.
6. While waiting for the conversion of the analog data, the digital computer feeds the first set of steering commands to the digital-to-analog converters. The converters then convert these commands into analog voltages.
7. At the completion of the analog-to-digital conversion, the binary data are fed into the digital computer.

8. The digital trajectory and guidance computations are then continued, using the new analog-sampled data. At the completion of these calculations, the digital computer waits for the next synchronizing pulse. The procedure, starting with item 5, is then repeated.

9. When the missile has attained the proper velocity magnitude, a "stop" command is transmitted from the digital computer. This command is used to halt the train of synchronizing pulses and to stop the analog computations.

Although the digital computer performs its calculations only when it has received new data, the analog computer performs its computations continuously, modified only by the periodic inputs from the digital computer.

6.5.18. The Digital Computer as a Function Generator. In actual practice, the atmosphere has a significant effect in the early portion of a ballistic missile's flight. In principle, the analog computer is well suited for studies of such factors as missile bending, wind gusts, and effects of aerodynamic forces; but realistic simulation requires a large number of function generators and/or multipliers. *Generation of the relevant functions by a digital computer* can increase accuracy and ease setup and checkout problems (Fig. 6.5.9*c*).

References

1. Susskind, Alfred (ed.): *Analog-Digital Conversion Techniques*, Technology Press, MIT, Cambridge, Mass., 1957.
2. Ford, Lester R.: *Differential Equations*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1955.
3. Hildebrand, F. B.: *Introduction to Numerical Analysis*, McGraw-Hill Book Company, Inc., New York, 1956.
4. Linvill, William K., and John M. Salzer: Analysis of Control Systems Involving Digital Computers, *Proc. IRE*, **41** (July, 1954).
5. Nielsen, Kaj L.: *Methods in Numerical Analysis*, The Macmillan Company, New York, 1956.
6. Salzer, Herbert E.: New Formulas for Facilitating Osculatory Interpolation, *J. Research Natl. Bur. Standards*, **52**(4):211–216 (April, 1954).
7. Salzer, Herbert E.: Osculatory Extrapolation and a New Method for the Numerical Integration of Differential Equations, *J. Franklin Inst.*, **262**(2):111–119 (August, 1956).
8. Salzer, John M.: Frequency Analysis of Digital Computers Operating in Real Time, *Proc. IRE*, **42** (February, 1954).
9. Leger, R. M., and J. L. Greenstein: *Control Eng.*, September, 1956.
10. Horwitz, R. D.: Testing Systems by Combined Analog and Digital Simulation, *Control Eng.*, September, 1959.

PART 6: MISCELLANEOUS TECHNIQUES

By L. D. Kovach

SOLUTION OF DIFFERENCE EQUATIONS

6.6.1. Difference Equations. A technique for solving difference equations with the electronic difference analyzer has been described by Wadel.[1] The general linear difference equation with constant coefficients can be written as

$$\sum_{k=0}^{n} a_k \mathrm{E}^k x(t) = f(t) \tag{6.6.1}$$

where E is an operator defined by

$$\mathrm{E}x(t) = x(t+h) \qquad \mathrm{E}^2x(t) = x(t+2h) \qquad \text{etc.} \tag{6.6.2}$$

It will be assumed that the original problem statement has been normalized to render $h = 1$.

In Eq. (1) the a_k are given constants, and $f(t)$ is specified at equidistant values of the independent variable t separated by $h = 1$. It is required to find $x(t)$ for the discrete series of t values such that Eq. (1) is satisfied.

A simple numerical example of general equation (1) is

$$\mathrm{E}^3x + 0.6\mathrm{E}^2x + 0.4\mathrm{E}x + 0.5x = 20 \tag{6.6.3}$$

given $x(0) = 0$, $x(1) = +5$, and $x(2) = -5$. The solution of Eq. (3) is found by classical means (e.g., Pipes[2]) to be

$$x = 17.5(0.773)^t \sin(1.43t + 2.94) - 11.6(-0.836)^t + 8.0 \quad (t = 0, 1, 2, \ldots) \tag{6.6.4}$$

In solving the difference equation (3) on the differential analyzer, integrators are not used as such, but as "memory" or "storage" elements. Figure 6.6.1 indicates

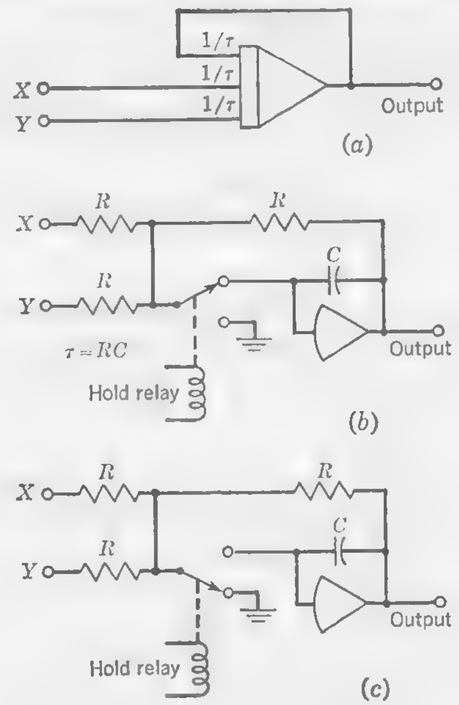


FIG. 6.6.1. D-C integrator used as an analog storage device: patchbay connections (*a*), OPERATE condition (*b*), and HOLD condition (*c*).

the external feedback connection to be used for all the integrators. An integrator so connected will be called a "store," and has the following properties:

Mode of Integrator	*Action of Store*
RESET	Output stationary at the "initial-condition" value
OPERATE	Output approaches a value equal to the negative sum of all the inputs (excluding input from its own output) with a time constant $\tau = 0.1$ sec (τ must be small compared with h)
HOLD	Output stationary at the value existing at moment of executing HOLD

The state of an integrator is caused to be RESET, OPERATE, or HOLD by the computer reset and hold relays (Sec. 4.1.). In the present application, the hold relays are divorced from the main computer control system and are controlled by a stepping relay in the manner detailed below.

The computer circuit diagram for the solution of Eq. (3) is shown in Fig. 6.6.2. The states of all stores are controlled according to the following scheme:

State of stores	Arabic-numbered	Roman-numbered
1	HOLD	OPERATE
2	OPERATE	HOLD

This two-state sequence is repeated as many times as necessary to cover the desired range of the independent variable, an increment $h = 1$ of the independent variable being obtained for each repetition. For the investigations described in this section 1 sec of computer time was used for each state, so that 2 sec correspond to each increment of the independent variable.

Before the computer system is placed in state 1, the arabic-numbered stores are RESET and initial conditions are established corresponding to the given values of $x(0)$, $x(1)$, and $x(2)$, as indicated in Fig. 6.6.2.

When the system is placed in state 1, then the arabic-numbered stores HOLD their output at their respective initial conditions; the roman-numbered stores OPERATE, and therefore their outputs assume values equal to the negatives of their respective inputs. Thus, at the end of state 1, the output of store 1 is $-x(2)$, that of store II is

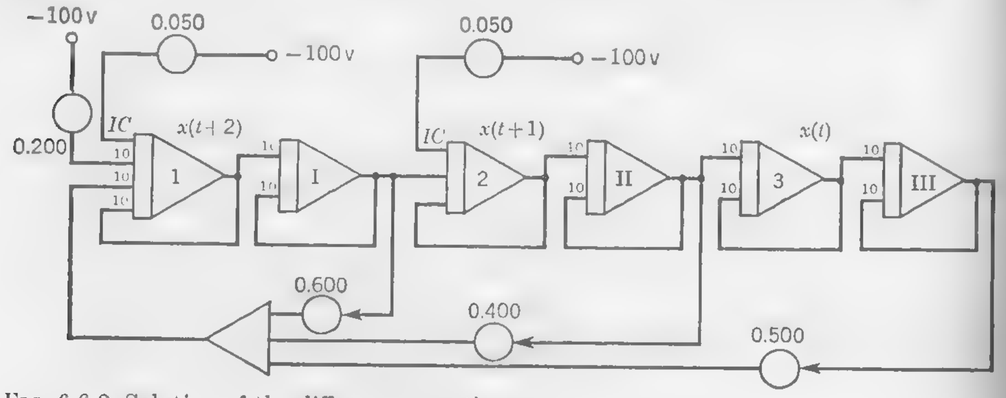


FIG. 6.6.2. Solution of the difference equation

$$E^3x + 0.6E^2x + 0.4Ex + 0.5x = 20$$

$-x(1)$, and that of store III is $-x(0)$. When the system passes to state 2, the roman-numbered stores HOLD these values as outputs.

During state 2, the arabic-numbered stores OPERATE: the output of store 1 assumes the value $20 - 0.5x(0) - 0.4x(1) - 0.6x(2)$, and by Eq. (3) this sum equals $x(3)$; the output of store 2 assumes the value $x(2)$; and the output of store 3 assumes the value $x(1)$. Thus, at the close of state 2 the consecutive values $x(1)$, $x(2)$, and $x(3)$ have replaced the initially set-in values $x(0)$, $x(1)$, and $x(2)$. The computer is next returned to state 1 for another cycle of operation. As these two-state cycles are carried out, the output of store 3 successively takes on the values $x(0)$, $x(1)$, $x(2)$, $x(3)$, $x(4)$, . . . representing the desired solution $x(t)$ of Eq. (3).

While only linear difference equations (with constant coefficients) have been considered, nonlinearities may be introduced by means of the usual analog-computing elements.

COMPUTATION OF FOURIER INTEGRALS WITHOUT ANALOG MULTIPLIERS*

6.6.2. A Weighting-function Technique. Given a suitable function $f(t)$ *different from zero only for* $t > 0$, straightforward analog computation of the Fourier transform

* The editor has taken the liberty of adding some more recent material, based mainly on Ref. 5, to Dr. Kovach's manuscript.

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}\,dt = A(\omega) - jB(\omega)$$
$$A(\omega) = \int_0^{\infty} f(t)\cos\omega t\,dt \qquad B(\omega) = \int_0^{\infty} f(t)\sin\omega t\,dt \tag{6.6.5}$$

for a set of values of ω requires analog multiplication as well as integration.[3] The following, more elegant method[4,5] requires only linear operations. The simple differential equation

$$-\frac{1}{\omega}\frac{d^2y}{dt^2} - \omega y = f(t) \tag{6.6.6}$$

has the weighting function (response to a unit impulse at $t = \tau$)

$$W(t-\tau) = -\sin\omega(t-\tau) \qquad (t > \tau) \tag{6.6.7}$$

The solution $y(t)$ for $y(0) = y'(0) = 0$, which is easily generated by the simple analog-computer setup of Fig. 6.6.3, is given by the superposition integral

$$y(t) = \int_0^t f(\tau)W(t-\tau)\,d\tau = -\sin\omega t\int_0^t f(\tau)\cos\omega\tau\,d\tau + \cos\omega t\int_0^t f(\tau)\sin\omega\tau\,d\tau \tag{6.6.8}$$

If the Fourier transform (5) exists, the two integrals on the right converge, respectively, to $A(\omega)$ and $B(\omega)$. Hence *the amplitude* a *and the phase angle* φ *of the resulting steady-state sinusoidal oscillation*

$$y_{ss}(t) = -a\sin(\omega t + \varphi) \tag{6.6.9}$$

are precisely the absolute value $|F(\omega)|$ *and the argument* arg $F(\omega)$ *of the desired complex*

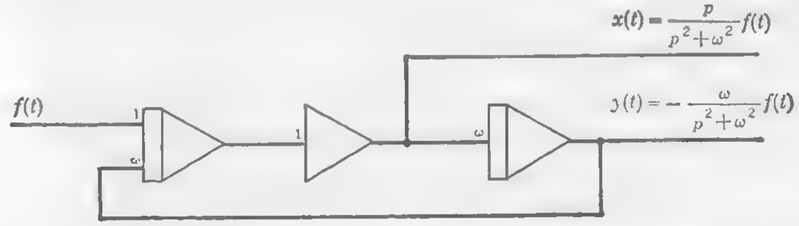


FIG. 6.6.3. Fourier analysis without analog multipliers.

Fourier transform $F(\omega)$. For convenience, the computer setup of Fig. 6.6.3 also produces

$$x(t) = -\frac{1}{\omega}\frac{dy}{dt} = \cos\omega t\int_0^t f(\tau)\cos\omega\tau\,d\tau + \sin\omega t\int_0^t f(\tau)\sin\omega\tau\,d\tau \tag{6.6.10}$$

which converges to the steady-state solution

$$x_{ss}(t) = a\cos(\omega t + \varphi) \tag{6.6.11}$$

Hence both $A(\omega) = |F(\omega)|\cos\varphi$ and $B(\omega) = -|F(\omega)|\sin\varphi$ can be read out at the time $t = nT = 2\pi n/\omega$ where $n = 1, 2, \ldots$ must be sufficiently large to yield steady-state conditions within the required accuracy. If desired, an appropriate timing circuit (perhaps a second sine-generating loop) could place the computer in HOLD for digital read-out; or an inverse resolver may be used to read a and φ directly.

6.6.3. Inverse Fourier Transforms.[4] Given the Fourier transform (5) of a real function $f(t)$, one can obtain the separate terms of the inverse Fourier transform

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\tau)e^{\pm j\lambda\tau}\,d\tau = \frac{1}{\pi}\int_0^{\infty} A(\tau)\cos\lambda\tau\,d\tau \pm \frac{1}{\pi}\int_0^{\infty} B(\tau)\sin\lambda\tau\,d\tau \tag{6.6.12}$$

by computing each integral on the right in the manner of Art. 6.6.2.

6.6.4. Harmonic Analysis of Periodic Functions.[4,5] The computer setup of Fig. 6.6.3 also yields the coefficients

$$a_k = \int_0^{T_0} f(t) \cos k\omega_0 t \, dt \qquad b_k = \int_0^{T_0} f(t) \sin k\omega_0 t \, dt \tag{6.6.13}$$

in the Fourier expansion

$$\frac{2\pi}{\omega_0} f(t) = \frac{1}{2} a_0 + \sum_{k=1}^{\infty} (a_k \cos k\omega_0 t + b_k \sin k\omega_0 t) \tag{6.6.14}$$

of a suitable *periodic* input $f(t)$. It is only necessary to read the output voltages (8) and (10) at the time $T_0 = 2\pi/\omega_0$ with $\omega = k\omega_0$.

CONFORMAL MAPPING

6.6.5. Theory. Let the function $w = w(z)$ define a mapping of the complex z plane onto the complex w plane. As the point $z = x + jy$ ranges over any configuration in the z plane, the point $w = u + jv$ describes a corresponding figure in the w plane.

6.6.6. An Airfoil Transformation.[6] As an example, a circle

$$x = x_0 + R \cos \omega t \qquad y = y_0 + R \sin \omega t \tag{6.6.15}$$

is transformed into an airfoil shape by the transformation

$$w = z + \frac{a}{z} + \frac{b}{z^2} + \frac{c}{z^3} \tag{6.6.16}$$

One generates the functions (15) of the time t by a sine-generating loop and uses further computing elements to obtain

$$\begin{aligned} u &= x + \frac{ax}{x^2 + y^2} + \frac{b(x^2 - y^2)}{(x^2 + y^2)^2} + \frac{c(x^3 - 3xy^2)}{(x^2 + y^2)^3} \\ v &= y - \frac{ay}{x^2 + y^2} - \frac{2bxy}{(x^2 + y^2)^2} + \frac{c(y^3 - 3x^2y)}{(x^2 + y^2)^3} \end{aligned} \tag{6.6.17}$$

with a scale factor of 1 volt per unit. Different airfoil shapes may be matched by different choices of the constants a, b, c, x_0, and y_0.

References

1. Wadel, L. B.: An Electronic Differential Analyzer as a Difference Analyzer, *J. Assoc. Computing Machinery*, **1**(3):128–136 (July, 1954).
2. Pipes, L. A.: *Applied Mathematics for Engineers and Physicists*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1958.
3. MacNee, A. B.: A High-speed Product Integrator, *MIT-Research Lab. Electronics Rept.* 136, August, 1949.
4. McCool, W. A.: Frequency Analysis by Electronic Analog Methods, *Project Cyclone Symposium I*, pp. 95–101, Reeves Instrument Corp., New York, March, 1951.
5. Teasdale, A. R.: Computing Fourier Coefficients without the Use of Multipliers, Temco Aircraft Corp., Dallas, Tex., 1959.
6. Tomlinson, N. P., *et al.*: Analog-computer Construction of Conformal Maps in Fluid Dynamics, *J. Appl. Phys.*, **21**:307 (1950).

Section 7

TRANSISTORIZED ELECTRONIC ANALOG COMPUTERS

By

HERMANN SCHMID, *Senior Design Engineer, Link Division of General Precision, Inc., Binghamton, N.Y.*

WALTER HOCHWALD, *Supervisor, Inertial Navigation Department, Autonetics Division, North American Aviation, Inc., Downey, Calif.*

HAROLD L. EHLERS, *Supervisor, Flight Control Section, Autonetics Division, North American Aviation, Inc., Downey, Calif.*

CONTENTS

PART 1: TRANSISTOR ANALOG-COMPUTING ELEMENTS

Hermann Schmid

Introduction

7.1.1. Transistor vs. Vacuum-tube Computing Elements 7–2

Operational Amplifiers and Switches

7.1.2. Transistor D-C Amplifiers ... 7–2
7.1.3. Linear Operations 7–6
7.1.4. Transistor Voltage Switches .. 7–6
7.1.5. The Diode-ring Modulator ... 7–9

Multiplier Circuits

7.1.6. Introduction 7–9
7.1.7. Transistor Time-division Multipliers 7–9
7.1.8. Transistor Triangle-integrating Multipliers 7–14
7.1.9. The Transistor Binary-increment Multiplier 7–15
7.1.10. Hall-effect Multipliers and Magnetoresistor Multipliers .. 7–18
7.1.11. Division with Transistor Feedback Multipliers 7–18

Function-generator Circuits

7.1.12. Introduction 7–20
7.1.13. Transistor Bidirectional Limiters 7–20
7.1.14. Inverse-trigonometric-function Generators 7–21
7.1.15. Trigonometric-function Generators 7–23
7.1.16. Square-root Generators 7–25
7.1.17. Electronic Rectangular-to-polar Coordinate Conversion Unit 7–26

PART 2: DESIGN OF A HIGH-QUALITY TRANSISTOR D-C AMPLIFIER

Walter Hochwald and Harold L. Ehlers

Requirements and Basic Theory

7.2.1. Requirements 7–29
7.2.2. Amplifier Parameters 7–29
7.2.3. Amplifier Configurations 7–30

Circuit Design

7.2.4. Chopper Amplifier 7–32
7.2.5. Integrating Amplifier 7–34
7.2.6. A-C Preamplifier 7–35
7.2.7. Output Power Amplifier 7–35
7.2.8. Specifications and Test Data .. 7–35

PART 1: TRANSISTOR ANALOG-COMPUTING ELEMENTS*

By HERMANN SCHMID

INTRODUCTION

7.1.1. Transistor vs. Vacuum-tube Computing Elements.[1] As compared with their vacuum-tube counterparts, transistor computing elements

1. Can make electronic analog computers smaller and lighter, especially in mobile and air-borne installations
2. Consume less power, thus permitting smaller low-voltage power supplies, which, in turn, might also use semiconductor components
3. Dissipate less heat, thus minimizing the need for ventilation or cooling
4. Have longer life and thus higher reliability (no filament or cathode failures)
5. Are easier and cheaper to manufacture, since transistors adapt themselves much better to etched-circuit-board construction and dip soldering
6. Perform better in many cases; the saturation characteristic of the transistors permits the design of excellent voltage limiters and voltage switches

On the other hand, computing elements employing germanium transistors

1. Require ambient temperatures between 0 and 50°C; substitution of silicon transistors may raise the upper limit to 100°C
2. Impose limitations on the computing voltages (±10 to ±50 volts) and hence on the signal-to-noise ratios obtainable; this situation is improving as high-voltage transistors become available
3. Increase initial costs of equipment somewhat; but the cost of transistors is decreasing consistently as production increases

In general, most analog-computing elements using vacuum tubes can also be built with transistors; only circuits in which the use of transistors improves or changes the performance of the circuit, or produces certain problems, will be discussed. In addition, transistors permit the design of several computing circuits[2] which are impossible or impractical with any other electronic component.

OPERATIONAL AMPLIFIERS AND SWITCHES

7.1.2. Transistor D-C Amplifiers[3,4] **(See also Arts. 7.2.1 to 7.2.8).** Table 7.1.1 lists the relative advantages and disadvantages of vacuum-tube and transistor d-c amplifiers. In transistor d-c amplifiers, drift is a particularly severe problem; in addition to sporadic changes in transistor characteristics one has to contend with variations due to temperature changes. Drift-compensation methods are similar to those used with vacuum-tube d-c amplifiers (Sec. 2.3); in addition, there are special temperature-compensation techniques.[5,6] A second serious problem is posed by the low input impedance of transistor amplifiers and by the variations of the inherent voltage offset of the base-to-emitter junction. Special feedback networks and compensating circuits can alleviate these input problems.[7,8]

Acceptable accuracy and small drift can be obtained from transistor d-c amplifiers only with automatic balancing techniques (Sec. 2.3). Out of the large number of modulators and demodulators used for stabilization purposes, the following groups and types can be distinguished.

* The author and editor are grateful to Link Aviation, Inc., for their permission to publish the circuits of Figs. 7.1.1 to 7.1.21. Link Aviation, Inc., has patents or patents pending for a number of these circuits.

Electromechanical	Semiconductors	Others
Vibrating reed[1] Commutator[1]	Diode half wave[9] Diode ring[9]	Photo-diode/light[27] Second-harmonic[28] magnetic
Mercury-wetted vibrating reed	Switching transistor[7,11,13]	Variable capacitor[29]

Table 7.1.1. Advantages and Disadvantages of Vacuum-tube vs. Transistor D-C Operational Amplifier

Type	Advantages	Disadvantages
Vacuum-tube d-c amplifiers	Large output voltage range High input impedance No interstage loading Large power output without severe bandwidth restriction	Large size High power dissipation (heaters) Finite life (tubes) Limited reliability Sensitive to shock Tubes are microphonic Drift in tube parameters
Transistor d-c amplifiers	Small size, light weight Small power dissipation No heaters Long life High reliability Insensitive to shock	Temperature-sensitive Small output voltage range Low input impedance Interstage loading Drifts in transistor parameters Small power output, requires buffer Small bandwidth with buffer

Table 7.1.2. Comparison between Electromechanical and Semiconductor Modulators

Type	Advantages	Disadvantages
Electromechanical modulators	Not sensitive to temperature Practically infinite open-circuit impedance Closed-circuit resistance practically zero No voltage of current offset No requirements on the source of the signal to be switched	Short life (except for mercury-wetted switch) High cost Large in size, heavy Poor reliability Sensitive to shock
Semiconductor modulators	Small size, light weight High switching frequency Long life High reliability Insensitive to shock Small power dissipation	Sensitive to temperature Has voltage and current offset Finite open-circuit impedance Finite closed-circuit impedance Requires low source impedance Small output voltage range

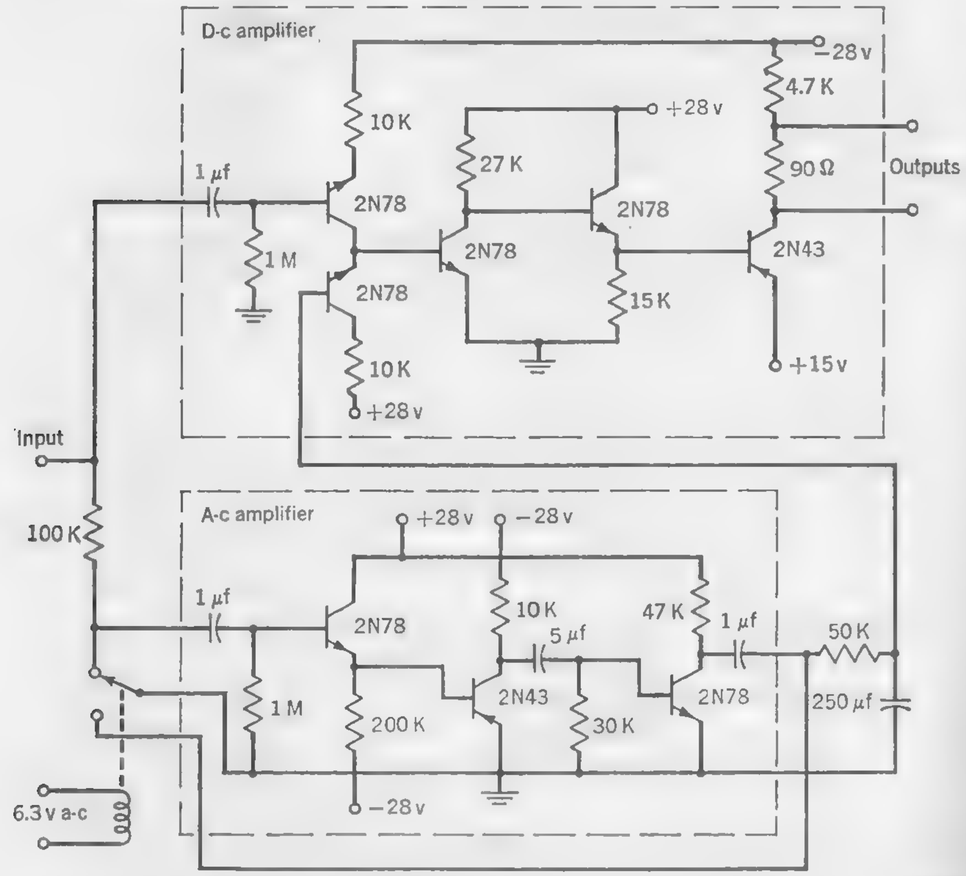


FIG. 7.1.1. Simplified transistor d-c amplifier with electromechanical chopper for modulation and demodulation. (Biasing and phase-shifting networks are not shown.)

The advantages and disadvantages of electromechanical vs. semiconductor modulators are illustrated in Table 7.1.2. This comparison applies also to demodulators, but their operation is much less critical. It is obvious that electronic, or, more specifically, semiconductor modulators would be preferred to electromechanical types if they would perform equally well; unfortunately, this is in general not true.

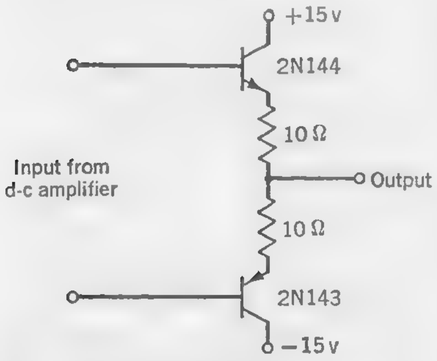


FIG. 7.1.2. One-watt complementary-symmetry output stage.

The transistor d-c amplifier shown in Fig. 7.1.1 employs an electromechanical chopper for both modulation and demodulation. The chopping frequency is 60 cps but with proper circuit changes it might equally well employ a 400-cps chopper. The power output of the amplifier shown is less than 50 mw; if more power is required, a power amplifier has to be added. One version of such a power amplifier, a complementary-symmetry output stage, is shown in Fig. 7.1.2. The power output can thus be boosted to approximately ¼ watt in free air, and with an appropriate heat sink to 1 watt. Power transistors have very low alpha cutoff frequencies, and the addition of the buffer amplifier would limit the bandwidth of the amplifier considerably. Fortunately, these buffer amplifiers are seldom required. Other characteristics of the

amplifier (without the buffer stage) are:

Output voltage range ±10 volts
Maximum drift (referred to input) ±1 mv between 20 and 50°C
Maximum total noise 1 mv peak to peak
Open-loop gain $>10^6$ for $f < 2$ cps
Open-loop gain $>2 \times 10^4$ for 2 cps $< f <$ 20 cps

The amplifier illustrated in Fig. 7.1.3 is a combination of individual circuits presented in a paper by G. B. B. Chaplin and A. R. Owens.[7] It employs one transistor for the modulator and another for the demodulator. In order to cut the current drift of the transistor modulator to a minimum, the reverse bias across the base-to-emitter

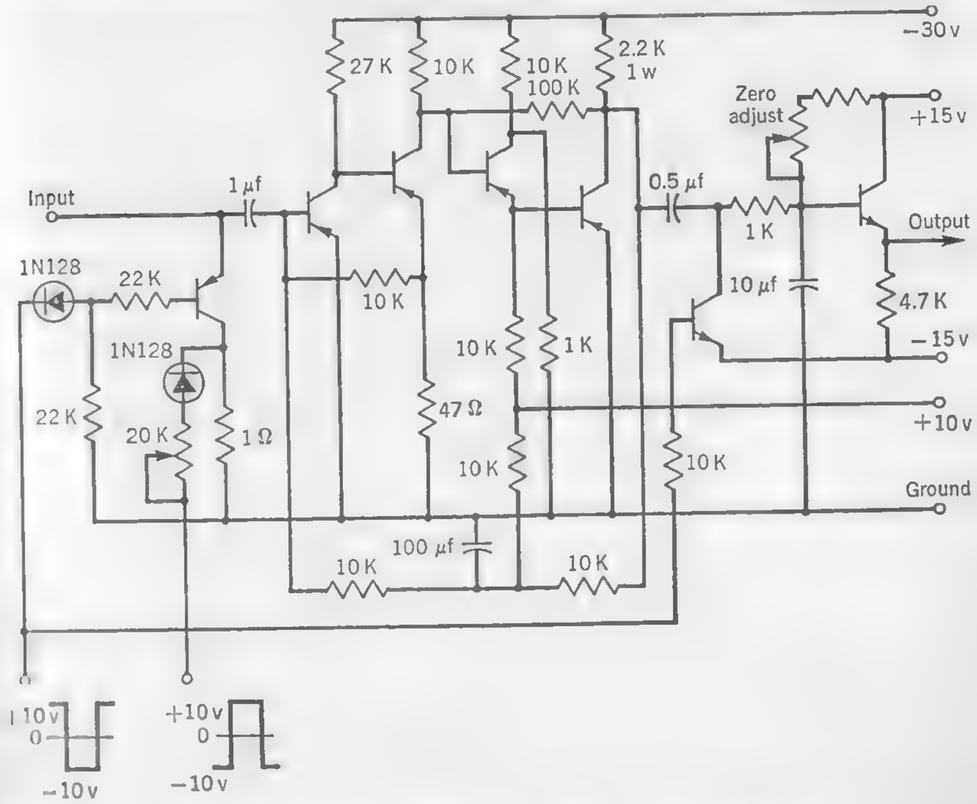


FIG. 7.1.3. Complete schematic of transistor d-c operational amplifier, with a transistor modulator and demodulator. (*G. B. B. Chaplin and A. R. Owens, United Kingdom Atomic Energy Association.*)

junction of this transistor is made zero. This is achieved by placing a diode in the base lead of the modulator transistor. For minimum drift this transistor should be of the low I_{co} type such as surface-barrier or silicon-alloy junction transistors. The chopping frequency for this amplifier may vary widely; practical limits are 400 cps to 10 kc. The power and bandwidth restrictions of this amplifier are similar to those listed for the circuit in Fig. 7.1.1. One serious disadvantage of this type of carrier amplifier is that it cannot be used as an integrating circuit, since any capacitance between input and output would make the circuit unstable.

With a carrier frequency of 1.6 kc, the following performance data are specified[7] for the circuit without the output stage:

Output voltage range is ±10 volts.
Current drift referred to input is 4×10^{-9} amp (between 20 and 50°C).
Voltage drift referred to input is 100 μv (between 20 and 50°C).
Transfer open-loop gain is 50 volts per μa.

Instead of the transistor modulator and demodulator in Fig. 7.1.3, a diode modulator and demodulator can be used with the same or a similar amplifier circuit as illustrated in Fig. 7.1.4. The half-wave modulator shown there can be replaced by a diode ring modulator[9] illustrated in Fig. 7.1.5*d*; several other demodulators are also possible. The stability of such a d-c amplifier depends, of course, on the stability of the forward and inverse diode characteristics. Special silicon diodes with extremely low inverse voltage must be used in these modulators. Several manufacturers are presently marketing specially selected and matched diodes, four in one unit, for this particular application.

7.1.3. Linear Operations. Addition, subtraction, integration, and differentiation with transistor d-c amplifiers are basically the same as with vacuum-tube d-c amplifiers (Sec. 2.2). Besides the advantages and disadvantages listed for transistor analog-computing elements, there are, however, the general problems which exist with any transistor circuit. One of these is the low output voltage range (± 10 to ± 50 volts) which decreases the signal-to-noise ratio and thus requires more extensive and more careful shielding of input and output leads. Another common problem exists in the contact resistance of the connector of plug-in units. When large signals are used, contact resistance is of little consequence, but when the signal is in the millivolt region, the problem becomes much more severe. Transistor analog-computing elements thus require better connectors with lower contact resistance.

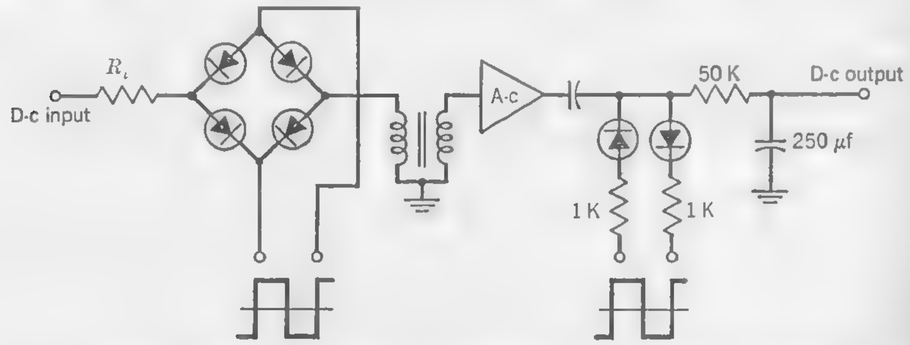


FIG. 7.1.4. Simplified diagram of a transistor d-c amplifier using a diode-bridge modulator and a two-diode demodulator.

The scaling resistors of d-c operational amplifiers are limited by the input impedance of the amplifier to a certain maximum value, approximately 100 kilohms. Therefore, the input impedance of an adding circuit is lower than is used for vacuum-tube amplifiers, which in turn requires lower source impedances of the driver circuits (potentiometers) if loading errors are to be avoided.

For integrating circuits, lower scaling resistors mean larger values of capacitors for the same time constant. Since, however, the voltages are also lower, the physical size of the capacitors is not increased.

7.1.4. Transistor Voltage Switches. The saturation and cutoff characteristics of many types of junction transistors,[10,11] such as the germanium-alloy junction transistors, are exceptionally suitable for the purpose of switching voltages (i.e., for connecting either one of the two voltages at the input terminals to the output terminal) by means of an appropriate base control signal. The operation of such a transistor switch, which consists of two transistors connected emitter to emitter, is similar to that of a single-pole double-throw relay.

The circuits of the push-pull transistor voltage switch and of the complementary transistor voltage switch are illustrated in Fig. 7.1.5*a* and *b*, respectively. When the control voltage on the base of one transistor, say T_1, is "high" and the control voltage on the base of T_2 is "low," T_1 is cut off and T_2 is saturated, and thus the potential on the collector of T_2 appears at the output. When the control voltages are reversed, T_1 saturates and T_2 is cut off, and thus the potential on the collector of T_1 appears at the output.

A transistor *saturates* when both its base-to-emitter and base-to-collector junctions are forward-biased. The voltage across the saturated transistor V_s is a measure of the switching quality. For most germanium-alloy junction transistors, V_s is of the order of 1 mv when the load impedance is larger than 100 kilohms and the base control current is between 0.3 and 5 ma.

A transistor is *cut off* when both its base-to-emitter and base-to-collector junctions are reverse-biased. The current through the cutoff transistor is also a measure of the performance, but only with respect to its current-switching quality.

The maximum signal to be switched with these transistor devices is limited by the breakdown potentials of the switching transistor;[12] for a transistor with breakdown voltages of 25 volts or higher between any two electrodes, the maximum signal to be switched is 20 volts of one polarity or ± 10 volts if both polarities are required.

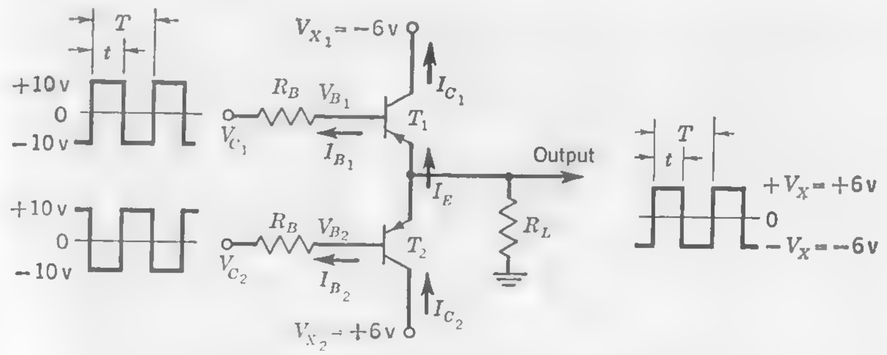


FIG. 7.1.5*a*. Push-pull transistor voltage switch.

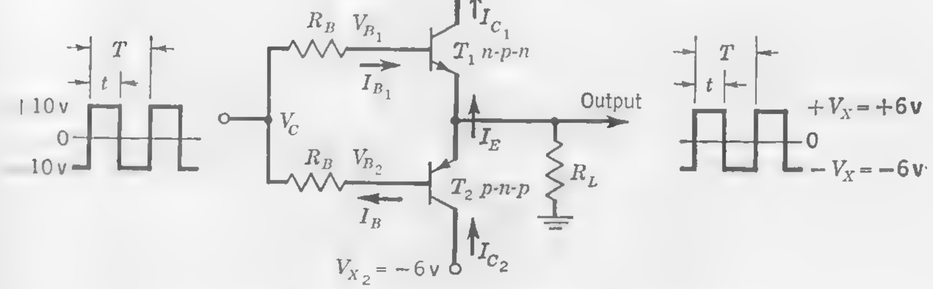


FIG. 7.1.5*b*. Complementary transistor voltage switch.

The switching transistor required not only must have low saturation voltage and high breakdown voltage, but it also must operate rapidly. The most important single factor determining the speed of switching is the alpha cutoff frequency f_{α_c}. Transistors with a high f_{α_c} generally have at least one low breakdown voltage, usually V_{BE}; transistors with high V_{BE} have generally a low f_{α_c}. A compromise between the two opposing requirements indicates the choice of germanium-junction alloy transistors with breakdown voltages of 25 volts or larger and an f_{α_c} of 6 Mc. With the appropriate amount of base current and suitable input circuitry, rise and fall times of the order of 0.1 μsec can be obtained.

The base current of the saturated transistor flows through the collector into the source of the signal to be switched. Any impedance in this source would change the potential of the signal to be switched, and it is therefore necessary that this impedance be as small as possible. A practical value is the output impedance of an operational amplifier, i.e., less than 1 ohm.

The current through the load resistor R_L will flow through the saturated transistor,

reduce the emitter to base current, and increase the saturation voltage. It is therefore important that R_L be as large as possible.

These transistor switches do not require any balancing or zero setting. With a saturation voltage of 1 mv and a maximum signal of 10 volts, the dynamic range is of the order of 10,000. There is very little difference between the performance of the push-pull and the complementary transistor voltage switch. The only difference consists in the emitter current I_E which flows during the transient or switching time.

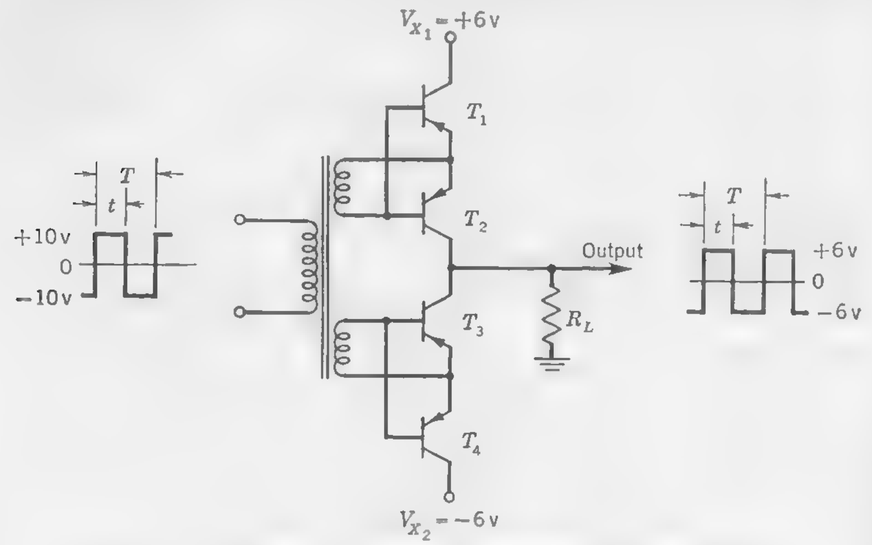


FIG. 7.1.5c. Transformer-coupled four-transistor voltage switch.

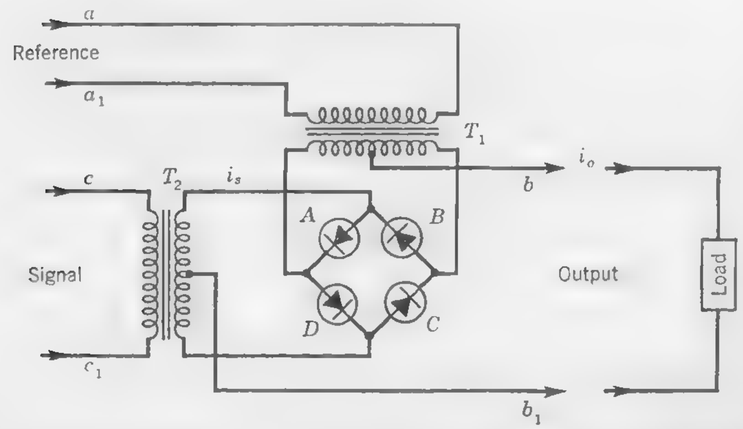


FIG. 7.1.5d. Diode-ring modulator. (*Courtesy of P. M. Thompson, Defence Research Communication Establishment, Ottawa, Canada.*)

For the push-pull switch, I_E is of the order of 1 ma for both polarities of the input signal, while for the complementary switch I_E may be as low as 20 μa when V_{x1} is positive and as high as 10 ma when V_{x1} is negative. This large transient current requires, of course, a smaller source impedance of V_x. However, since these currents flow only during the short transient time (approximately 0.1 μsec), i.e., the time when the transistors change from one state to another, the problem is not so serious as might appear. It is a further advantage of the push-pull switch that it uses two identical transistors, but a disadvantage that it requires push-pull control signals. Similarly, the single-ended base control signal is an advantage of the complementary switch, while the requirement of one *n-p-n* and one *p-n-p* transistor is a disadvantage.

Besides the transistor voltage switches described, there are a number of different types of transistor switches which will operate and perform similarly[11,13] but which have some undesirable limitations. The best of these is the transformer-coupled four-transistor voltage switch illustrated in Fig. 7.1.5*c*. The push-pull control signals are coupled to the bases by means of an input transformer, whereby one end of each secondary winding is returned to the emitter junctions. This biases the base control signals to the potentials present at the emitters and eliminates the need for transistors with a large emitter-to-base breakdown voltage. Transistors with higher f_{α_e} can be used in this switch, which is therefore considerably faster than the types described previously. The saturation voltage V_s will be of the order of 2 mv because two transistors, connected in series, are saturated at any one time. The disadvantage of this switch is that because of the transformer coupling the base control signals must have a constant duty ratio. The switch can therefore not be used in time-division circuits.

7.1.5. The Diode-ring Modulator.[14] The diode-ring modulator is shown in Fig. 7.1.5*d* with three sets of input terminals, a-a_1, b-b_1, c-c_1. Any two of these sets may be used as input terminals, while the remaining set will be the output terminals.

Some of the classical operating modes are as follows:

Use as a Phase-sensitive Rectifier. Two sinusoidal waveforms of the same frequency can be applied to a-a_1 and c-c_1. The device acts then as a phase-sensitive rectifier and produces a direct current at terminals b-b_1. The amplitude of this direct current depends, in particular, on the relative phase of the two input waveforms. For square-wave inputs, the magnitude of the direct current is a linear function of this phase difference.

Use as a Suppressed Carrier Modulator. For two input waveforms of differing frequency applied to a-a_1 and c-c_1, the output at b-b_1 is the instantaneous product of the input waveforms. This is a suppressed carrier modulation.

Current Operation of the Diode-ring Modulator. Suppose that the input terminals in Fig. 7.1.5*d* are supplied from a constant-current (infinite-impedance) source a-a_1 with a square wave (reference) and b-b_1 with a smaller current (signal). The output is then the current flowing through the short-circuited terminals b-b_1.

The current reference wave i_R will drive one rectifier pair AB (or CD) into conduction and the alternate pair CD (or AB) will be biased to cutoff by the forward voltage drop generated across the conducting pair. It follows that any signal current i_S injected into transformer T_2 can reach the output terminals only via one of the conducting diode pairs. Determined by the reference, the rectifiers perform a simple switching action, whereby the current from either the upper or lower half of T_2 is selected. This switching action is undisturbed by the amplitude and polarity of i_S, as long as $i_S < i_R$. The reverse currents of the cutoff diodes constitute an error which produces unbalance. With silicon diodes, the percentage error can be made very small. In addition, losses in the transformers will disturb the perfection of the diode bridge; however, the error so introduced is also very small.

The balance in actual diode-ring modulators is such that the zero-signal output current is of the order 10^{-3} of the full-scale output current.

MULTIPLIER CIRCUITS

7.1.6. Introduction. Several transistor multipliers make use of the excellent voltage-switching characteristic of the new transistor switches. In spite of the voltage limitations of these switches, better signal-to-noise ratios are possible than with most other electronic switches; the transistor voltage switch can also switch potentials of both polarities, thus permitting four-quadrant multiplications.

7.1.7. Transistor Time-division Multipliers. The operation of the electronic time-division multiplier (Sec. 3.2) is based on the principle that the area under a rectangular pulse, which has been subjected to pulse-width and pulse-amplitude modulation, is proportional to the product of the two modulation variables. With vacuum-tube circuitry, the pulse-width modulation can be performed very accurately. Accurate amplitude modulation involves fairly complex vacuum-tube switching circuits, which

may require biasing or push-pull operations for four-quadrant multiplication; here the use of transistor voltage switches (Art. 7.1.4) constitutes a real advantage.

The performance of any time-division multiplier[1] (Fig. 7.1.6*a*) is always, at best, a compromise between static and dynamic accuracy. Assuming that maximum-bandwidth components are already used, the choice of the pulse repetition rate will determine whether higher static and lower dynamic performance (with low carrier frequency) or lower static and higher dynamic performance will be obtained. In the above assumption, the bandwidth of the filter was also assumed maximum. Maximum bandwidth with minimum phase shift can be obtained from a filter only when it has a large number of "zeros," which means a large number of components, usually including large and heavy inductors. Since such filters would be highly undesirable in transistor circuitry, another compromise between the physical size and the dynamic

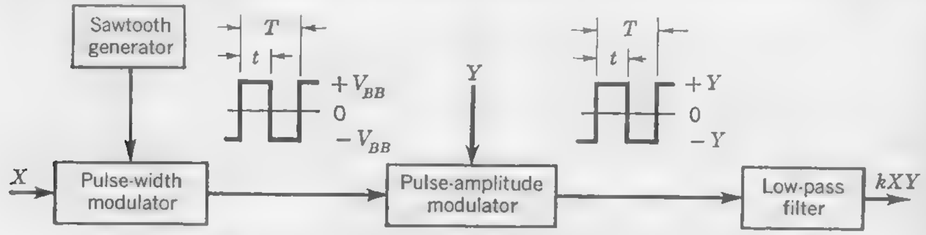


FIG. 7.1.6*a*. Block diagram of an open-loop time-division multiplier.

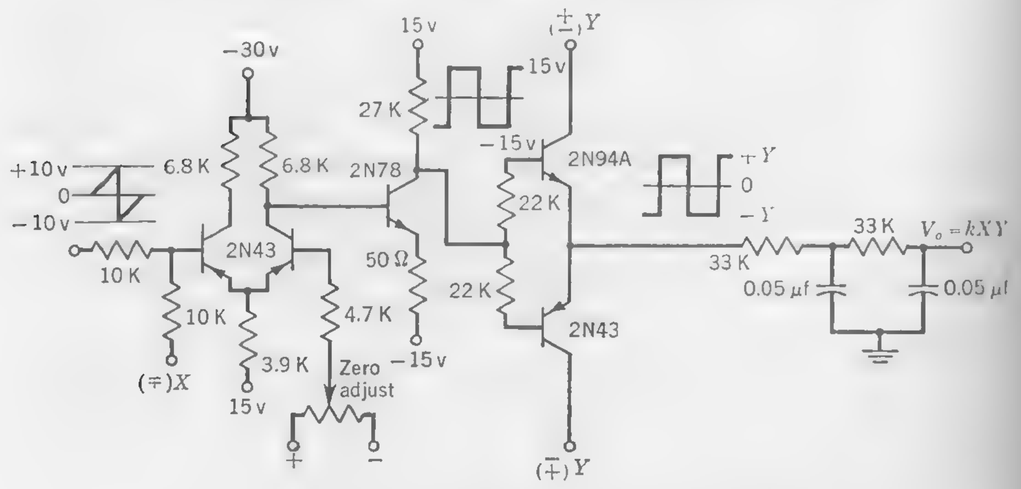


FIG. 7.1.6*b*. Open-loop transistor time-division multiplier.

performance has to be made. With a three-section *RC* filter, the 3-db bandwidth is 50 cps for a 1-kc carrier frequency.

The open-loop transistor time-division multiplier of Fig. 7.1.6*b* consists of a three-transistor pulse-width modulator and a complementary-symmetry transistor voltage switch. The pulse-width modulation is achieved by biasing a linear and amplitude-stable sawtooth (or triangular) wave by the first multiplication variable $\pm X$ and amplifying a small slice of this biased wave (± 20 mv with respect to ground) into a train of rectangular pulses. The width of these pulses, which control the transistor voltage switch, is proportional to X. The amplitude of the pulses at the output of the transistor switch is equal to Y. After filtering, the output voltage is proportional to XY. When the sawtooth frequency is 1 kc or lower, the static performance of the multiplier is largely determined by the linearity and amplitude stability of the sawtooth wave and the drift in the squaring amplifier. The dynamic performance is a function of the carrier frequency and the filter used. The multiplier of Fig. 7.1.6

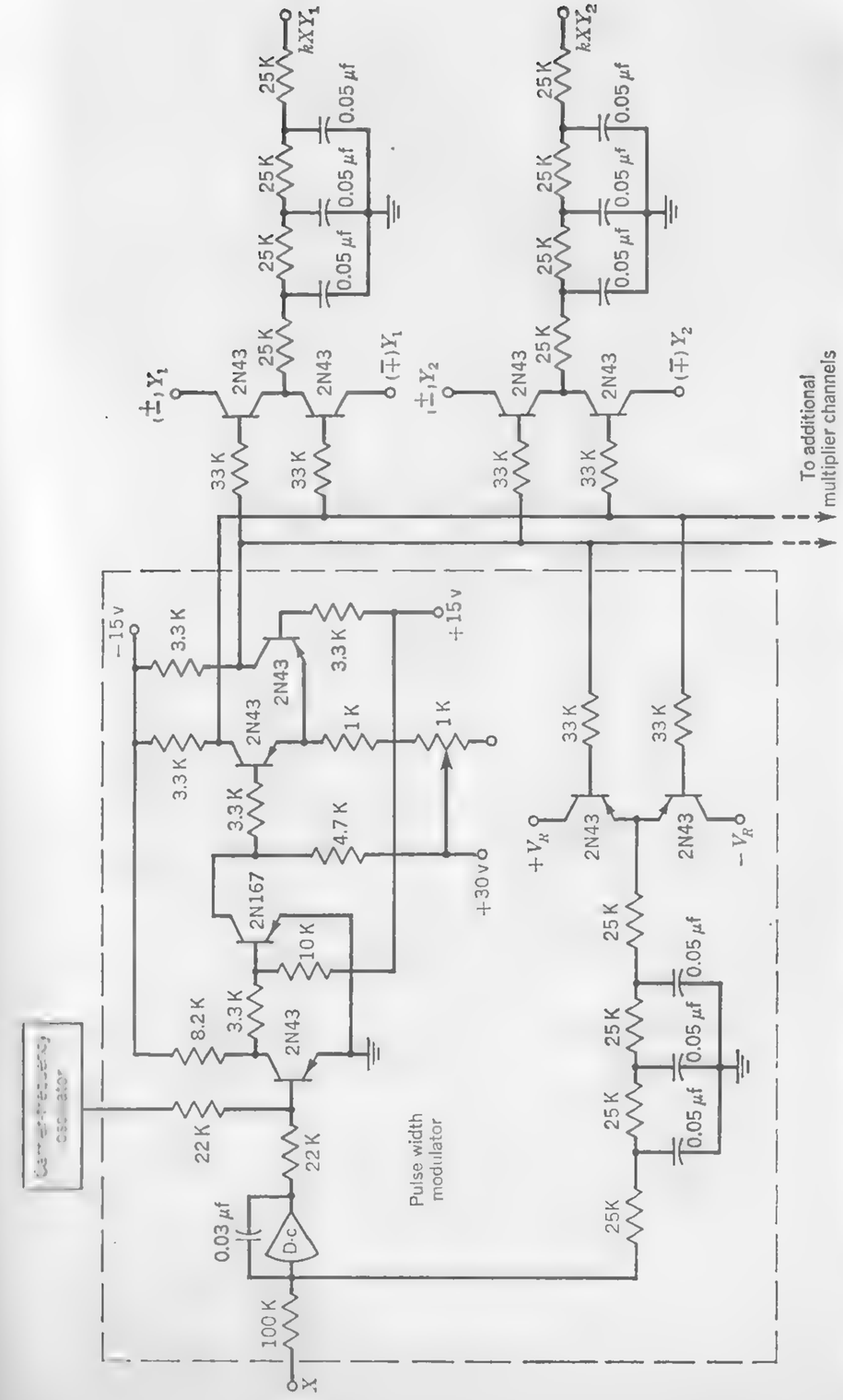


FIG. 7.1.7. Low-frequency version of a transistor feedback time-division multiplier for the multiplication of X by several Y variables.

will operate over all four quadrants and will maintain a static accuracy of better than ±1 per cent of full-scale output over a temperature range from 20 to 50°C. At any one temperature, the accuracy may be as high as ±0.2 per cent. With a 1-kc carrier and a three-section *RC* low-pass filter, the output voltage will drop by −3 db at 50 cps. Raising the carrier frequency to 10 kc will increase the 3-db bandwidth to 500 cps at some expense in static accuracy.

The *closed-loop time-division multiplier*[15] of Fig. 7.1.7 employs feedback-controlled pulse-width modulation in the manner of Sec. 3.2 for greatly improved performance. Unlike in Fig. 7.1.6*b*, the pulse-squaring amplifier of Fig. 7.1.7 has a push-pull output and drives push-pull transistor switches. The number of switches that can be driven is determined by the power and current capabilities as well as the frequency response of the squaring amplifier and the static and dynamic performance required from the multiplier. A practical maximum for an output stage with 150-mw transistors is six switches, thus providing multiplication of X by five variables of Y ($XY_1, XY_2, \ldots, XY_5$).

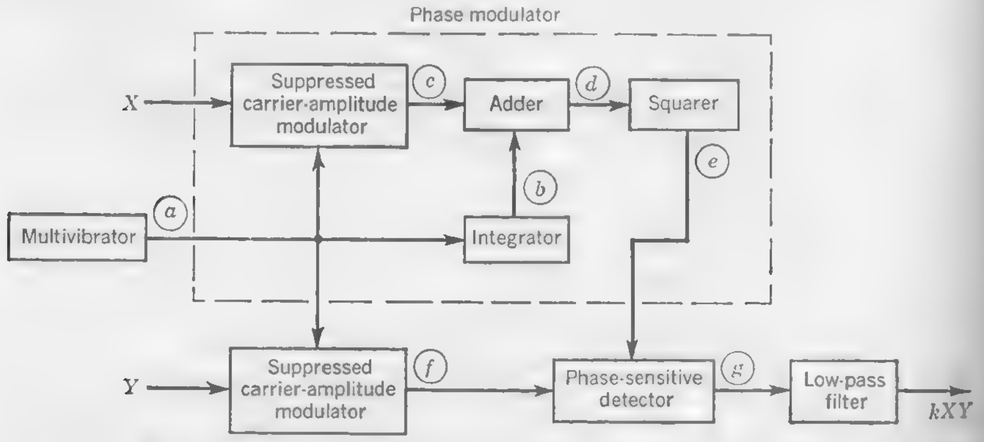


FIG. 7.1.8*a*. Block diagram of the transistor phase-amplitude multiplier. (*Courtesy of P. M. Thompson, Defence Research Communication Establishment, Ottawa, Canada.*)

The static accuracy of this transistor multiplier can be better than ±0.1 per cent of the full-scale output (±10 volts) over all four quadrants and within temperature variations of 25°C. In any quadrant, an accuracy better than ±0.05 per cent of full scale can be maintained.

With presently available switching transistors (breakdown voltages larger than 25 volts and $f_{\alpha_c} = 6$ Mc), a carrier frequency of 1 kc and a low-cost three-section *RC* filter, the output signal will be attenuated less than 3 db when either the X or the Y variable, but not both, is less than 50 cps. When both X and Y are functions of time, the 3-db bandwidth lies at 25 cps.

Transistor Multiplier Based on Simultaneous Phase and Amplitude Modulation.[4] In Fig. 7.1.8*a*, the two inputs X and Y (proportional to currents i_1 and i_2) respectively phase- and amplitude-modulate the square wave of frequency f_0. The amplitude-modulated square wave and the phase-modulated square wave comprise the signal and reference inputs of the phase-sensitive detector, whose output is proportional to the product of i_1 and i_2. Referring to Fig. 7.1.8*a* and to the waveforms of Fig. 7.1.9, the square wave (Fig. 7.1.9*a*) from the multivibrator is amplitude-modulated by a diode-bridge limiter (ring modulator, Art. 7.1.5). The resulting square wave of amplitude proportional X (Fig. 7.1.9*c*), is added to the integrator output voltage (Fig. 7.1.9*b*) to produce the waveform of Fig. 7.1.9*d*. Squaring and limiting yield the *phase-modulated* waveform of Fig. 7.1.9*e*. The phase-modulated voltage gates a second square wave, amplitude-modulated by the input current Y (Fig. 7.1.9*f*), to produce a pulse-width pulse-amplitude modulated output, like that of a time-division multiplier. This output is averaged to yield the desired product.

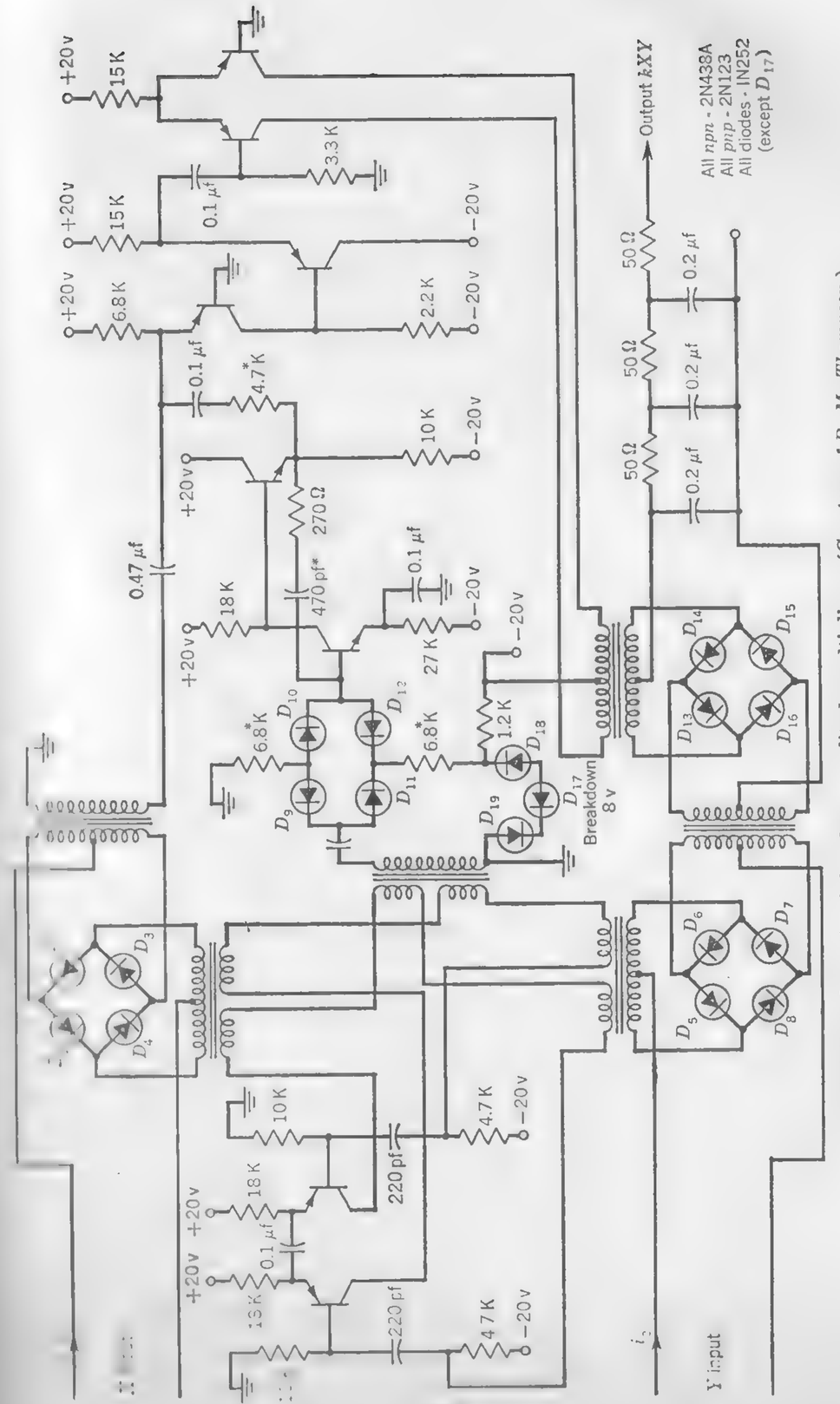


FIG. 7.1.8*b*. Schematic of the phase-amplitude multiplier. (*Courtesy of P. M. Thompson.*)

The performance of the multiplier depends on the precision with which the diode-ring modulators operate (Art. 7.1.5) and on the stability (freedom from drift) and linearity of the adder, squarer, and integrator. The components marked with an asterisk in Fig. 7.1.8*b* are especially sensitive with respect to characteristic or temperature changes. Since all transformers are required to pass the carrier square wave, they must pass relatively high frequencies and are thus wound on toroidal-tape permalloy cores. All windings connected to the diode rings are bifilar, while those connected to the multivibrator are separated to reduce interwinding capacitances.

The static accuracy of this multiplier is said to be $\pm\frac{1}{3}$ per cent of full-scale output (± 2 ma) over all four quadrants and within a temperature range from -50 to $+70$°C. It should be noted, however, that this circuit is designed to operate from high impedance into low impedance and that it will not function properly unless this is true.

With a carrier frequency of 50 kc, but without the filter, this accuracy can be maintained up to signal frequencies (X or Y) of 20 kc; however, when a three-section RC filter is used, the output is attenuated to -3 db at about 4 kc.

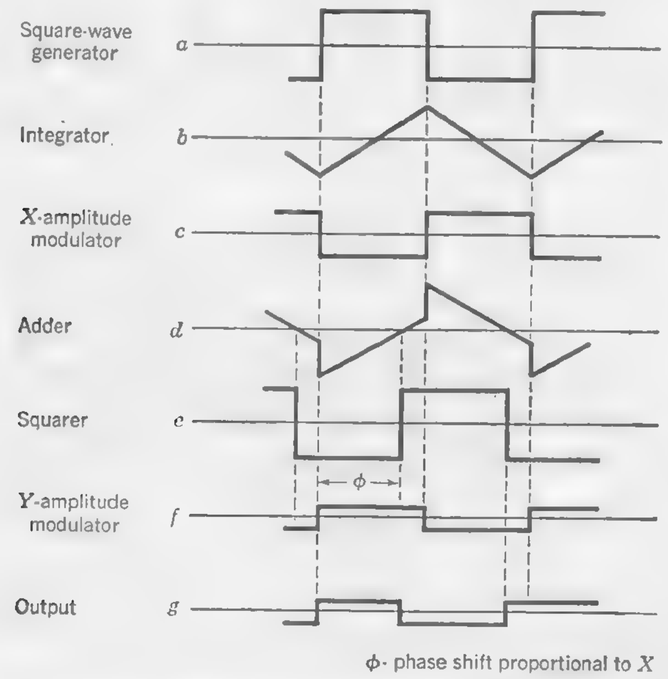


FIG. 7.1.9. Waveforms in the phase-amplitude multiplier. (*Courtesy of P. M. Thompson.*)

7.1.8. Transistor Triangle-integrating Multipliers. The operation of this multiplier[17] is based on the fact that the integral over a triangular wave, which has been biased to the first multiplication variable X and the excursions of which are limited to the second multiplication variable $+Y$ and $-Y$, is proportional to XY. The biasing of the linear and amplitude-stable triangular (or sawtooth) wave is accomplished by conventional adding circuits, while the clipping is performed by a transistor voltage switch. The simplest version of this multiplier, illustrated in Fig. 7.1.10, comprises only an input transformer, a complementary transistor switch, and a low-pass filter. The triangular wave is biased to the first multiplication variable by biasing the secondary of the input transformer to $\pm X$. The excursions of the triangular waves are limited accurately to the second multiplication variable by connecting the carrier wave to the bases and $+Y$ and $-Y$ to the collectors of the transistor voltage switch (Art. 7.1.4). Because the complementary switch can limit slowly varying a-c signals only when the potential on the collector of the n-p-n (or p-n-p) transistor is positive (negative), the multiplier will perform only over the two quadrants. The performance

of this multiplier depends critically on the linearity and amplitude stability of the triangular wave and on the temperature stability of the transistor switch. At a constant temperature, a static accuracy of ± 0.5 per cent of full scale or better can be obtained over the two quadrants. For a temperature variation of 25°C, ± 1 per cent static accuracy is more typical. Because the carrier wave has no sharp edges, considerably higher carrier frequencies can be used; so that a 3-db bandwidth as high as 5 kc is believed possible with the type of filter shown, still maintaining the static accuracy stated above.

7.1.9. The Transistor Binary-increment Multiplier. The principle of operation of this multiplier, which is an analog-digital hybrid, is that of a so-called step multiplier. Its basic elements are the analog-to-digital converter and a digitally controlled voltage divider[18] or variable-gain amplifier.

In Fig. 7.1.11*a*, the electromechanical switches of the step multiplier are replaced by transistor switches (Art. 7.1.4). These switches are driven from a reversible binary counter. The outputs from this counter are d-c voltages which are positive when the particular digit is ON and negative when the digit is OFF. Depending on the polarity of these outputs, the transistor voltage switches in the analog-to-digital converter loop will connect either the reference voltage V_R or ground to the scaling resistors. The switches in the variable-gain amplifier connect the Y variables or ground to the scaling resistors.

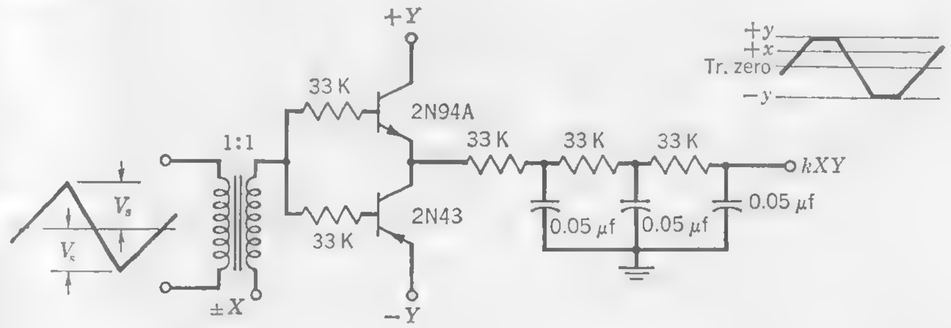


FIG. 7.1.10. Simple two-quadrant triangular-wave transistor limiter. The output of transistor limiter is a biased and clipped triangular wave.

The application of the transistor switches elevates the binary-increment multiplier from the very-low-frequency version (step multiplier) to a device that can multiply at frequencies of several kilocycles (Packard-Bell Multiverter M-2). But despite the excellent voltage-switching characteristics of the transistor switches, they are not as perfect as mechanical switches and thus produce certain design problems. Ultimately, however, not the transistor switches but the resistors limit the accuracy of this multiplier. One of the design problems is the fact that the resistance of the saturated transistor R_S has always a certain finite value (1 to 2 ohms) and that R_S is a function of the base current I_B. R_S is small when I_S is within certain limits ($\approx$1 to 10 ma). When the voltage to be switched and the base drive are constant, R_S can be compensated. For best accuracy, it is desirable to keep R_S as small as possible compared with the load resistance R_L. The range of values of the scaling resistors of the network shown in Fig. 7.1.11*a* is very large (from R to 2^nR). Such a wide variation of precision resistors is neither economical nor a practical load for the transistor switches. It is therefore much better to use a conventional ladder network.[19] The circuit in Fig. 7.1.11*b* is the binary-coded-decimal version of the original binary ladder network.

In the analog-to-digital converter of Fig. 7.1.11*a*, the gate will apply pulses to the reversible counter to increase the digital number when the output voltage V_o of the d-c amplifier is positive; when V_o is negative, it will decrease that number. Even if the gating frequency is very high, say 1 Mc, it would take 2^n μsec, or a maximum of approximately 16 msec for a 14-digit count. For this reason the so-called "put-and-

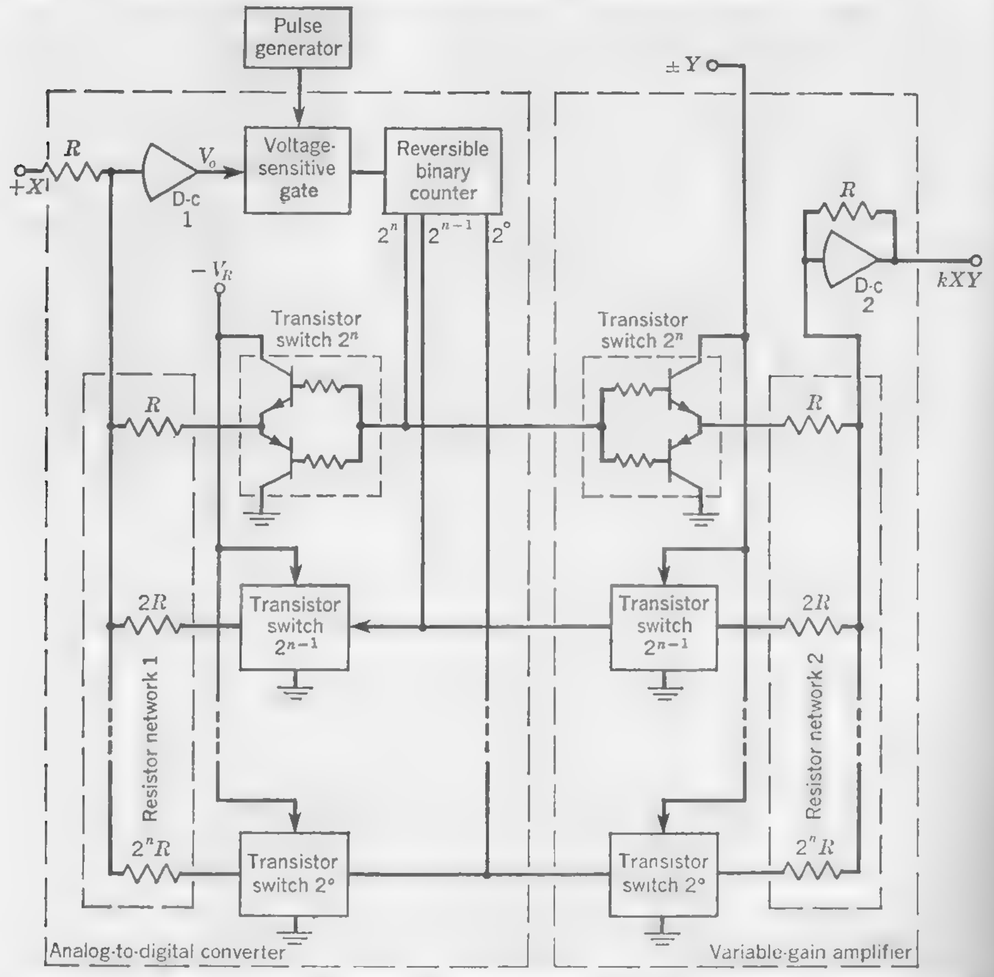


FIG. 7.1.11*a*. Simplified two-quadrant version of the binary-increment multiplier (step multiplier).

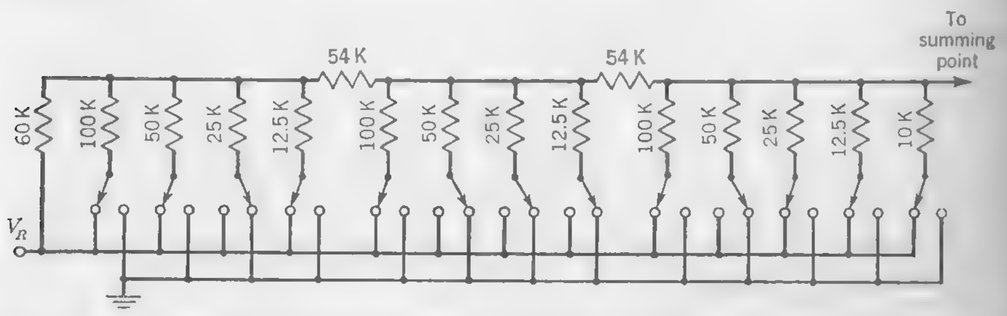


FIG. 7.1.11*b*. Binary-coded-decimal ladder network for the binary-increment multiplier. (*Courtesy of R. M. Beck, Packard-Bell Computer Corp.*)

take" method of successive approximation is used. This method operates at one digit at a time, starting with the most significant digit. Each time the system compares the magnitude of the input and feedback current. When the difference is below a certain minimum amount, the number presented by the counter circuit is the digital representation of the analog input X, exactly X/V_R. With this method of compari-

son, the time required for one conversion is sharply decreased from 2^n μsec to n times the time required for one comparison.

The time required for one conversion is determined by the bandwidth of the comparator circuit, by the setting time of the resistor matrix, and by the time required by the transistor switches and flip-flops to operate. Because high-gain chopper-stabilized d-c amplifiers have such a small bandwidth, a comparator circuit of the type shown in Fig. 7.1.11*c* is required. The significant points about this circuit are that it

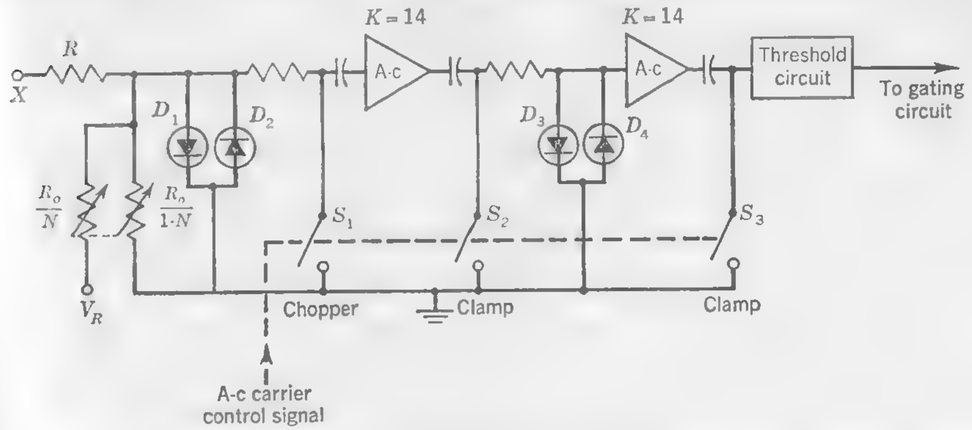


FIG. 7.1.11*c*. Simplified diagram of the comparison circuit for the binary-increment multiplier. (*Courtesy of R. M. Beck, Packard-Bell Computer Corp.*)

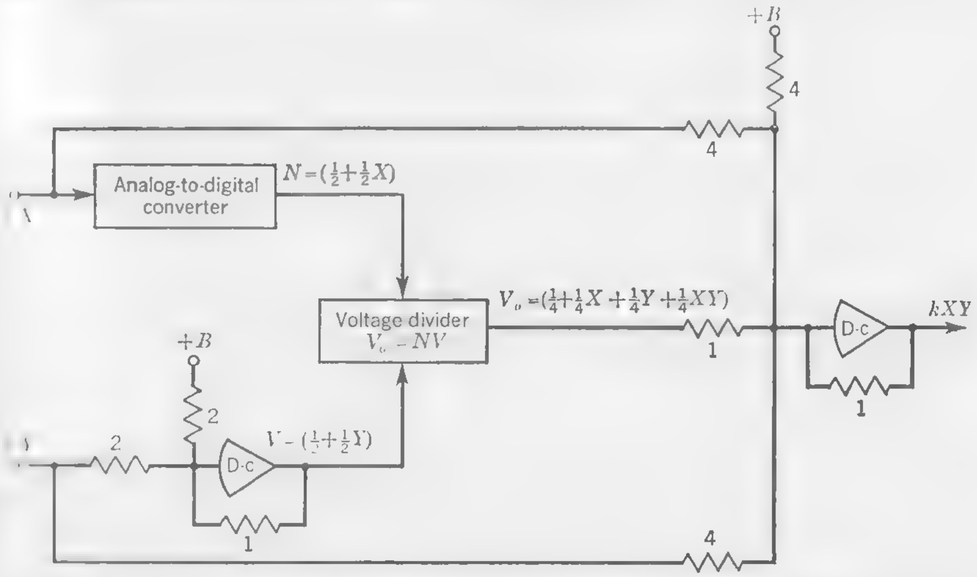


FIG. 7.1.11*d*. Four-quadrant version of the binary-increment transistor multiplier. (*Courtesy of R. M. Beck, Packard-Bell Computer Corp.*)

employs two transistor a-c amplifiers ($k = 14$) in series, a transistor chopper similar to that in Fig. 7.1.3, and two conventional clamping circuits. Only the clamping circuit with S_3 is required for the polarity detection. S_2 and the two diode networks are provided to prevent saturation of the amplifiers under any circumstances. The threshold circuit will provide zero input to the gating circuit when the input is smaller than a specified amount.

Four-quadrant operation with the binary-increment multiplier is achieved by the method illustrated in Fig. 7.1.11*d*.

The static accuracy obtainable from the binary-increment multiplier with 14 digits (Packard-Bell Multiverter M-2) is as high as ±0.04 per cent of the full-scale output over all four quadrants and over a temperature range of 15 to 45°C. The dynamic range of the device is, of course, $2^n = 2^{14} \sim 16{,}000$. The low-gain high-frequency a-c amplifier ($k \sim 200$) requires, however, a full-scale input current of 4 ma. With a d-c amplifier at the input, higher input impedances can be obtained, but usually not without sacrificing accuracy and frequency response.

The dynamic performance of the multiplier is determined by the frequency response of the analog-to-digital conversion loop, the bandwidth of the variable-gain d-c amplifier and that of any additional amplifiers used with the system. The frequency response of the analog-to-digital converter is said to be 4 μsec per bit, plus 4 μsec per conversion, which means 62 μsec for a 14-digit conversion.

7.1.10. Hall-effect Multipliers and Magnetoresistor Multipliers. Commercially available Hall generators consist of a thin wafer of semiconductor material such as indium arsenide or indium antimonide.[20] When it is placed into a magnetic field B and a control current I_c is pumped through the device at right angles to the magnetic field, a voltage appears at the mutually perpendicular direction. This Hall voltage V_H is proportional to the product kBI_c, where k depends on the Hall coefficient and on the dimension of the wafer.

The maximum Hall voltage V_H obtainable with commercially available generators is 0.5 volt even with a control current of 0.5 amp and a magnetic field of 10,000 gauss; input and output impedances of such a Hall generator are of the order of 1 ohm. Impedance levels and signal-to-noise ratio can be improved by an order of magnitude with Hall generators made of specially doped silicon.

An *open-loop Hall-effect multiplier* consists of a single Hall generator positioned in a magnetic field B, as shown in Fig. 7.1.12*a*. With the X variable converted into a magnetic field and the Y variable into a current, the output voltage from the Hall element is proportional to XY. Although the linearity and temperature stability of presently available Hall generators are limited, experimental multiplier models with special compensating circuits are said to have linearities of up to 1 per cent.[21]

Considerably better accuracy has been obtained with two identical Hall generators in a *feedback multiplier* circuit (Fig. 7.1.12).[22] Unfortunately, no two Hall generators are exactly alike. Lofgren[23] has obtained a static accuracy of 0.1 per cent of full scale, using a doped-silicon wafer; his paper discusses the details of one circuit, including a simple temperature-compensation method. It is also possible to compensate for the effects of asymmetrical contact placement.[24]

While Hall generators as such have an extremely wide bandwidth (up to 10^{12} sec^{-1}), the control-current driver amplifier and the electromagnet impose severe bandwidth restrictions. It is possible to trade flux density (and thus accuracy) for bandwidth by substitution of a ferrite core for the ferromagnetic core; thus Lofgren was able to trade 0.1 per cent static accuracy with 20 cps bandwidth for 0.3 per cent static accuracy with about 500 cps bandwidth.[23]

Magnetoresistance, an undesirable effect in the Hall multiplier element, is the change of resistance of a semiconductor in a variable magnetic field.[25] In magnetoresistors, this effect is specially emphasized, and the resistance is almost a linear function of the magnetic field. Magnetoresistors have been used in both open-loop and feedback multipliers.[22]

The linearity of a magnetoresistor is approximately ±3 per cent; the dynamic range is 10 for a change in magnetic field from 0 to 10 kilogauss. With higher magnetic fields, which are difficult to obtain, larger dynamic ranges can be obtained. The maximum impedance of present prototypes is 10 ohms with a magnetic field of 10 kilogauss. The minimum impedance with zero magnetic field is 1 ohm; however, higher impedances are said to be possible.

The bandwidth of the magnetoresistor alone is up to 10^{12} sec^{-1}; however, in practice it is limited by the frequency response of the electromagnet.

7.1.11. Division with Transistor Feedback Multipliers. Two-quadrant division by means of the transistor time-division multiplier of Fig. 7.1.7 does not differ greatly from division with a vacuum-tube time-division multiplier (Sec. 3.2). The transistor switch in the feedback path requires low source impedances of $+Z$ and $-Z$, so that these voltages must be obtained from d-c operational amplifiers. When the circuit in Fig. 7.1.7 is used for division, the ratio of the input and feedback resistor must be changed for optimum accuracy to 10:1, respectively. With this ratio of resistors, the circuit will saturate only when $X \leq 10\ Z$.

The static accuracy of this analog-division circuit is better than ±0.1 per cent of full scale when $Z > \frac{1}{10}$ of full scale, with a carrier frequency of 1 kc, and over a temperature range from 25 to 50°C. The accuracy is better than ±1 per cent when

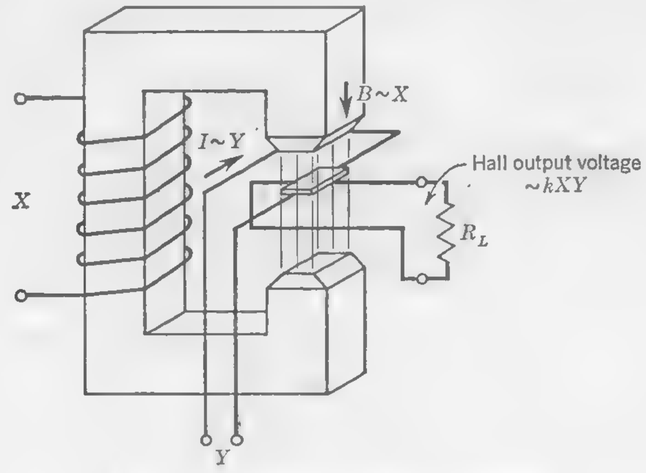


Fig. 7.1.12*a*. Open-loop Hall multiplier.

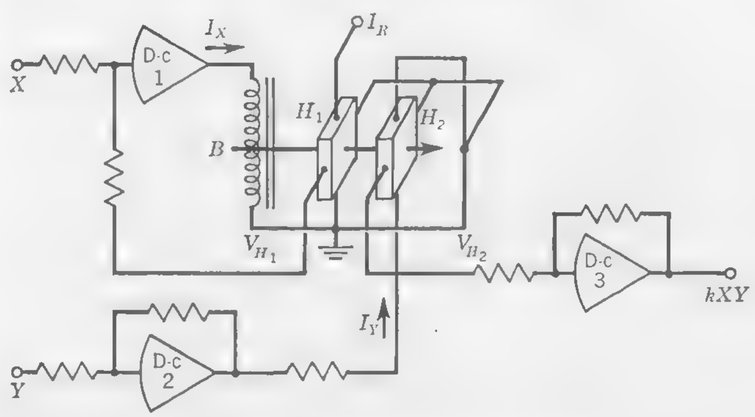


Fig. 7.1.12*b*. The feedback Hall multiplier.

$Z > \frac{1}{100}$ of full scale. Dynamic performance is essentially identical with that of the multiplier.

Two-quadrant division by means of the binary-increment multiplier of Fig. 7.1.11*a* is possible when the reference voltage is replaced by the Z variable. The quotient kX/Z appears in digital form. With the second voltage divider, in Fig. 7.1.11*a*, an output voltage of the form kXY/Z is obtained.

Feedback multipliers performing as analog-division circuits perform generally only over two quadrants. Most of these circuits can be made to accept also both polarities of the Z input variables through the addition of an absolute-value circuit (Sec. 3.3). However, such a circuit will *not* provide the correct output polarity if Z reverses its polarity.

A true four-quadrant voltage divider is the circuit in Fig. 7.1.13; it provides the

correct output polarity for any combination of input polarities. The transistor time-division multiplier will perform divisions over all four quadrants if the complementary transistor voltage switch S_3, inserted in the forward path of the feedback loop, reverses the polarity of the pulse-width modulated pulse trains to the transistor switches S_1 and S_2, when the polarity of Z changes. D-C amplifier 4 provides the required control potentials to S_3; +20 volts when $Z > +1$ mv; −20 volts when $Z < -1$ mv. The insertion of S_3 does not impede the performance of the device.

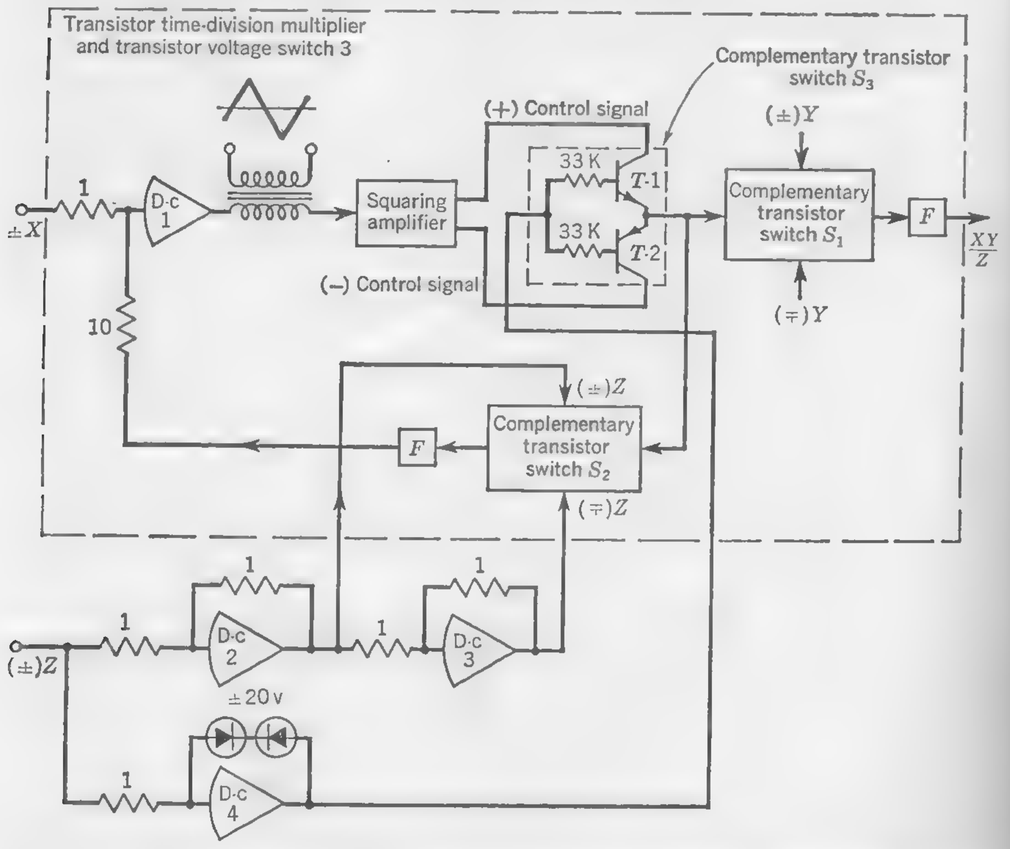


FIG. 7.1.13. A true four-quadrant dividing circuit with the transistor time-division multiplier.

All dividing circuits using feedback multipliers will provide the correct quotient for small divisor values only if the dividend is small. Special care must be taken to avoid instability in these cases.

FUNCTION-GENERATOR CIRCUITS

7.1.12. Introduction. Various transistor function generators for special functions combine small size and weight and low cost with remarkable performance. Most of these circuits are based on the excellent voltage-switching or limiting characteristics of transistor voltage switches; others convert functions of the time into function of an input voltage. Transistor limiters producing straight-line segment functions may also be combined into universal function generators just like diode limiters (Sec. 3.3).

7.1.13. Transistor Bidirectional Limiters. The simplest versions of the bidirectional limiter consist of only the complementary transistor voltage switch (Art. 7.1.4). It will limit any signal V_i connected to the bases exactly (within ±1 mv) to the voltages connected to the collectors $+V_L$ and $-V_L$. When $+V_L > V_i > -V_L$ and the

transistor voltage switch functions as emitter follower, V_o is approximately (within 0.1 volts) V_i, the difference being due to the base-to-emitter voltage V_{BE} and its variations with respect to temperature. For waveforms where the d-c component of V_o is of no importance, such as square waves, the complementary transistor voltage switch will perform as a high-quality bidirectional limiter. The low saturation voltage of germanium-alloy junction transistors results in excellent limiting characteristics, even for low limiting voltages.

Since, however, the d-c level of V_o is in most cases very significant, the complementary transistor voltage switch must be connected into the forward path of a low-drift d-c amplifier as illustrated in Fig. 7.1.14. In this connection, the output follows faithfully, with inverted sign, the input voltage V_i but only when $+V_L > V_i > -V_L$; in fact the complete circuit performs in this region like a d-c amplifier with an additional output emitter follower. When the limiter circuit is operating in the linear region, the d-c amplifier output voltage V_B differs from V_o only by approximately 100 mv (V_{BE}).

When the magnitude of V_i reaches and exceeds the magnitude of V_L, one of the two transistors will saturate. The magnitude of V_o does not follow V_i; the effect is the same as when the feedback resistor is increased. This will raise V_B more, which will saturate the transistor further. The final effect is that the feedback is interrupted,

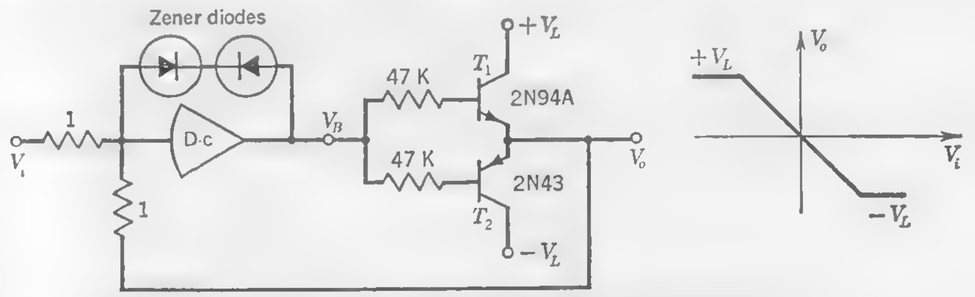


FIG. 7.1.14. Schematic of the high-accuracy bidirectional limiter.

V_B reaches its saturation value, and V_o is equal to V_L because one transistor is saturated and the other is cut off. This applies for both polarities of V_i. The polarity of V_L has to be such that the potential on the collector of T_1 is always positive with respect to the potential on the collector of T_2, irrespective of ground. It is not desirable to drive the d-c amplifier into saturation for two reasons: (1) V_B may be so high that the reverse voltages on the transistors are exceeded, and (2) chopper-stabilized d-c amplifiers require considerable time to return from saturation back to the linear region. The d-c amplifier may be kept out of saturation by connecting two Zener diodes (back to back) across it.

The slope of the V_o/V_i curve can be changed simply by changing the ratio between the input and the feedback resistor. The corners of this curve are defined precisely, because the circuit becomes regenerative when the magnitude of V_i reaches and exceeds the magnitude of V_L. Since the maximum value of V_L may be as high as 50 volts, the dynamic range of such a limiter is in the order of 5×10^4.

7.1.14. Inverse-trigonometric-function Generators. From the trigonometric relations on the sine wave in Fig. 7.1.15*a*, it can be seen that the zero crossings of a biased sine wave are t sec apart. If the sine wave is biased to a voltage $X = A_s \cos(\omega t/2)$, the time t between the zero crossings of the sine wave equals $\cos^{-1} X/A_s$. Since it is very simple to bias a sine wave, and fairly simple to measure the duration between the zero crossings, a function generator to solve $y = k \cos^{-1} X$ and $y = k \sin^{-1} X$ is not difficult to design.

The circuit shown in Fig. 7.1.15*b* solves the equation

$$V_o = ky \sin^{-1} \frac{X}{A_s} \tag{7.1.1}$$

The biasing is achieved by returning the secondary winding of transformer T_1 to the input variable X. The duration t between the zero crossings can be measured by amplifying a small slice of the biased sine wave (+20 mv with respect to ground) into a rectangular wave having a pulse width of t. A voltage analog to this time is produced by limiting the amplitude of the rectangular pulses to a certain value Y by means of a transistor voltage switch. As shown in Eq. (7.1.1), the value of Y, which may be either a constant or variable, multiplies the function.

The static accuracy of such a circuit is better than ±0.5 per cent of full scale, over an operating range from 15 to 50°C and better than 0.1 per cent at a constant temperature. To obtain this accuracy, it is required that the total harmonic distortion

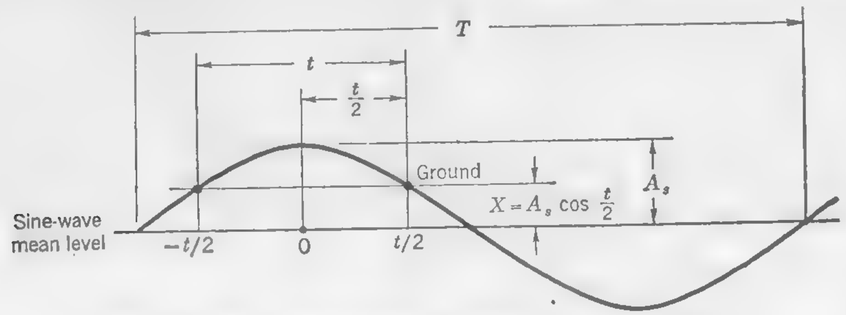


Fig. 7.1.15a. Trigonometric relations for the sine wave.

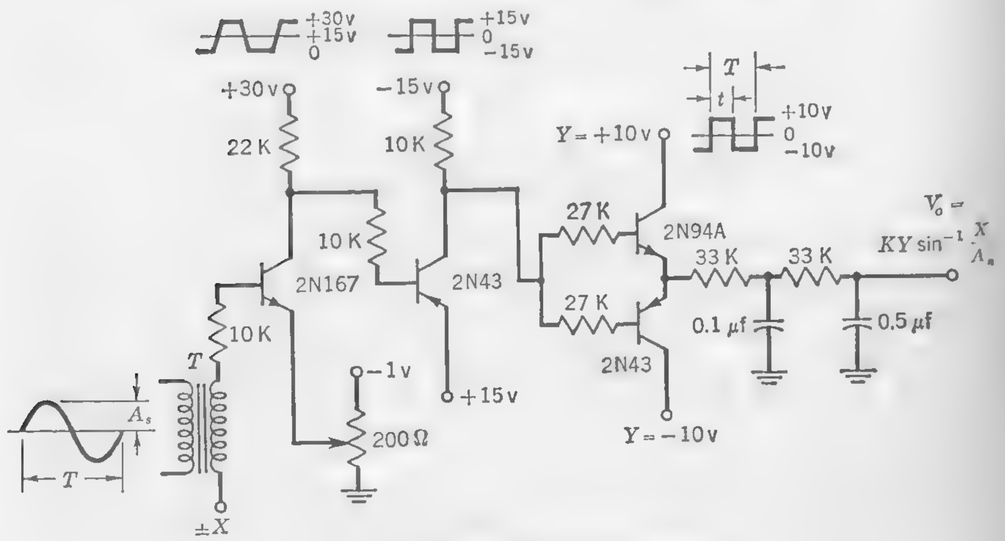


Fig. 7.1.15b. Arcsine-function generator.

of the sine wave be small (less than ±0.5 per cent) and that the amplitude of the sine wave A_s be stable and equal to the maximum excursion of X. The operating range of such a function generator is 180 deg. The temperature range can be considerably widened and the accuracy maintained when an emitter-coupled input stage is employed.

Since $\sin(\pi/2 - X) = \cos X$, one can operate the arcsine function generator as an arccosine function generator by adding the full-scale value to the negative output, so that

$$V_{90°} - V_o = ky \cos^{-1} \frac{X}{A_s} \tag{7.1.2}$$

The dynamic performance is essentially that of the open-loop time-division multiplier of Fig. 7.1.6b.

If y, the signal connected to the collectors of the complementary transistor voltage switch, is a sine wave from the same generator and also of constant amplitude, the output from the circuit in Fig. 7.1.15b becomes

$$V_o = k \cos\left(\sin^{-1} \frac{X}{A_s}\right) = k\left(1 - \frac{X^2}{A_s^2}\right)^{1/2} \tag{7.1.3}$$

where A_s is the constant sine-wave amplitude and k an arbitrary constant.

7.1.15. Trigonometric-function Generators. The operation of all-electronic trigonometric-function generators is based on the voltage-time relationship of the sine wave. A requirement for good performance of the sine and cosine function generators described below is therefore a pure (small harmonic distortion) and amplitude-stable sine wave. The input variable, which is a d-c voltage representing the angle, in general cannot be varied more than $\pm V_{90°}$.

Since $\sin X = \cos(\pi/2 - X)$ any sine-function generator may be converted to a cosine-function generator when the full-scale input voltage is added (with proper sign) to the input variable X.

Sine/Cosine Function Generator Based on Inversion. The function generator of Fig. 7.1.16 produces the inverse function of arcsine in the manner of Sec. 3.3.

The performance of this sine-function generator is almost completely determined by the performance of the arcsine generator, if a high-gain low-drift d-c amplifier is employed. The device may be made to solve the equation $V_o = k \cos X$ simply by

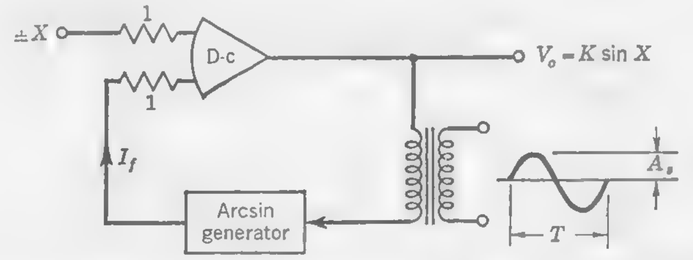


Fig. 7.1.16. Block diagram of sine-function generator based on the inversion principle.

adding the full-scale input voltage $V_{90°}$ to the input variable X. By itself this function generator may not appear to be very useful, but it should be noted that the pulse-width modulation circuit of the feedback time-division multiplier in Fig. 7.1.7 produces the sine function as a "by-product" when a sine wave is used as carrier. Using a sine wave as a carrier will not deteriorate the performance of the multiplier.[15]

Cosine/Sine Function Generator Based on the Integration of Sine-wave Portions.[26] Referring to Fig. 7.1.15a, the area under a portion of the sine curve equals the cosine of X, where X is the length of that portion; i.e.,

$$\cos X = -\int \sin X \, dX$$

One can implement this relation by generating rectangular pulses with a duration proportional to an input variable X. These pulses operate an electronic switch which connects only portions of a sine wave, of length X, to the input of an integrating network. The integrating network extracts the average from these sine-wave portions; this average is proportional to the cosine of X.

Since the pulses to the electronic switch must be in synchronism with the sine wave, one sine-wave generator supplies both the linear pulse-width modulator and the electronic switch. It has been shown[26] that the relation between the pulse width t and X is linear even when a sine wave is used as carrier. The electronic switch can be either the complementary or the push-pull transistor voltage switch (Art. 7.1.4). When only a single-ended signal is connected to the switch shown in Fig. 7.1.17, waveforms as illustrated in Fig. 7.1.18a appear at the output of the switch; when push-pull signals

are connected, waveforms appear as illustrated in Fig. 7.1.18*b*. With push-pull sine waves to the transistor switches, the function generator performs considerably better than with the single-ended sine wave, although the operation of the device remains the same. The integrating network is a low-pass filter.

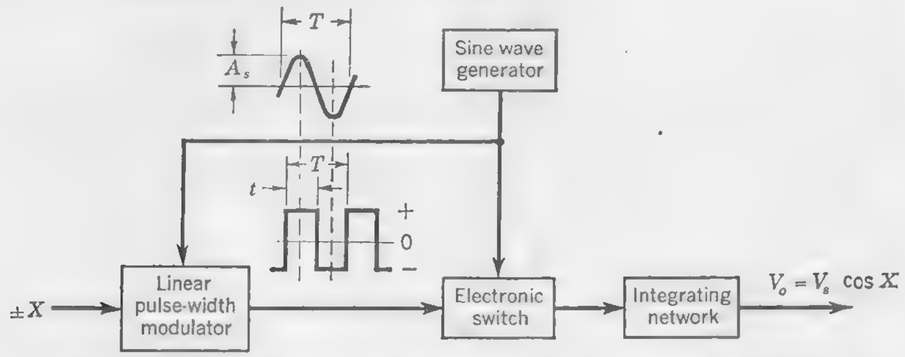


Fig. 7.1.17. Simplified block diagram of cosine/sine function generator.

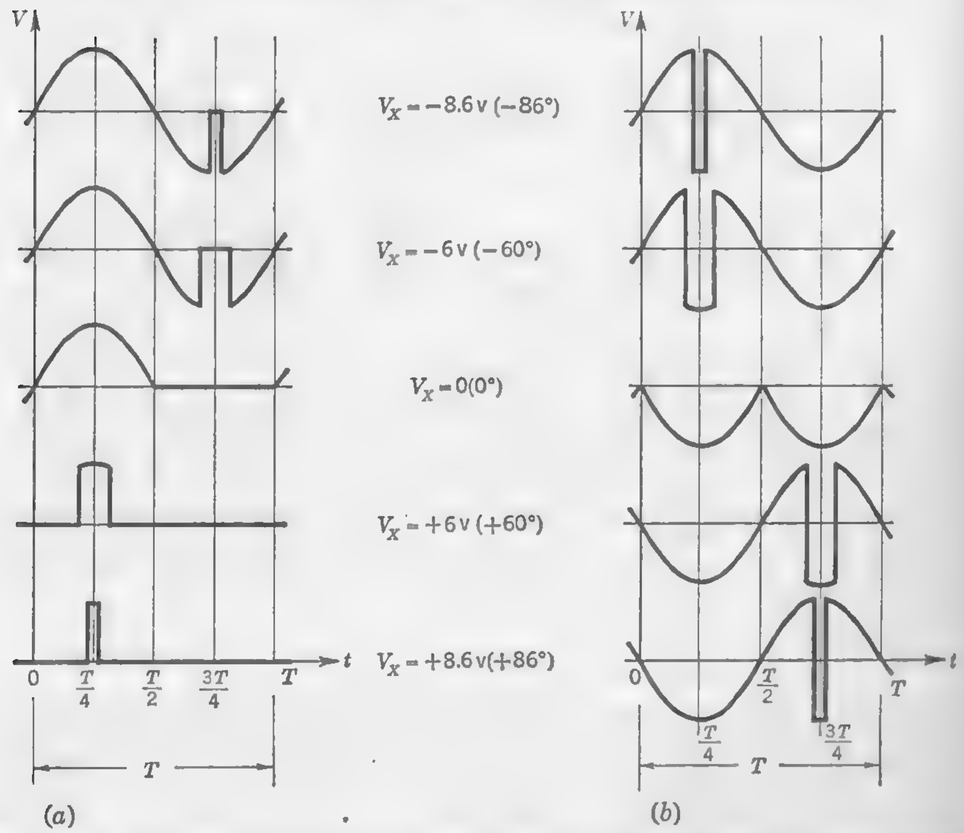


Fig. 7.1.18. Waveforms at the output of the switch of the cosine/sine function generator

The circuit of this cosine/sine function generator is essentially the same as that of the transistor time-division multiplier in Fig. 7.1.7. Only external connections have to be changed to convert this device from a multiplier to a trigonometric-function generator. The precision with which the sine wave can be switched depends on the synchronism of the rectangular pulses and the sine waves, on the rise and fall times of

the pulses, on the maximum sine-wave amplitude permissible, and on the saturation voltage of the switching transistors.

In order to have the sine wave and the rectangular pulses to the switch in synchronism, the phase shift introduced by the pulse-squaring amplifier on the rectangular pulses should be constant and as small as possible. Minimum phase shift as well as good pulse shape requires a wide-band pulse-squaring amplifier.

The static accuracy of such a cosine/sine function generator operating with a 1-kc carrier is better than ±0.1 per cent of the full-scale output, over an input range proportional to ±90 deg. Such accuracy implies a sine-wave amplitude stability of approximately 0.1 per cent, a maximum total distortion of less than 0.2 per cent, and a sine-generator source impedance of less than 1 ohm.

The dynamic performance of the cosine/sine function generator is similar to that of the transistor time-division multiplier of Fig. 7.1.7; the same compromises between static accuracy and dynamic performance, as well as filter performance and filter size, apply. With switching transistors having breakdown voltage of a minimum of 25 volts, the maximum output voltage is limited to ±5 volts.

The range of usable argument values of electronic sine-cosine generators may be shifted through addition of an input bias voltage; thus the cosine-function generator can be made to generate the sine function between zero and 180 deg.

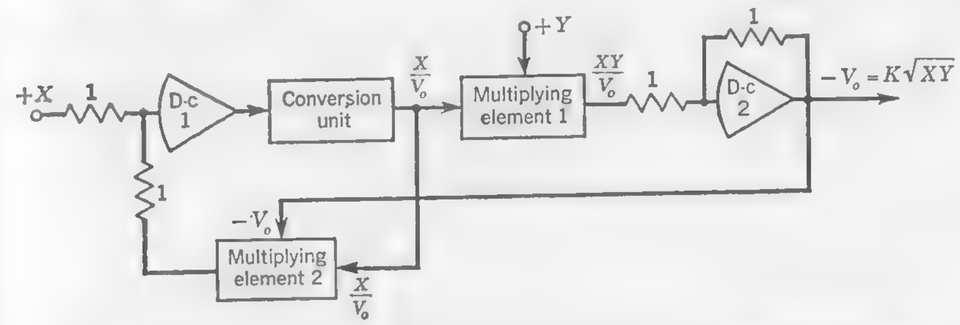


Fig. 7.1.19. Generation of $V_o = K\sqrt{XY}$ by means of a feedback multiplier.

The range of such a trigonometric-function generator can be further extended through all four quadrants when the polarity of the input signal or bias signal is reversed by an electronic switch (Sec. 3.3) or when the output from a positively biased sine-function generator and a negatively biased sine-function generator are summed.

7.1.16. Square-root Generators. A voltage proportional to the square root of an input voltage may be generated by inversion of the square function obtained with transistor multipliers or transistor triangle-integration squarers (Sec. 3.3). The circuit of Fig. 7.1.19, which can be built with any feedback multiplier, generates the square root of a product $\sqrt{XY}$ and has significant advantages over simple inversion circuits.

This square-root function generator is outstanding, especially, because its accuracy is independent of the magnitude of the input variable X and because it is completely stable when $X = 0$.

If the square-root generator in Fig. 7.1.19 is to be built with the transistor time-division multiplier of Fig. 7.1.7, the conversion unit will be the pulse-squaring amplifier and the multiplying elements the transistor voltage switches; if the binary-increment multiplier in Fig. 7.1.11*a* is to be used the conversion unit will be the voltage-sensitive gate and the reversible binary counter, while the multiplying units are the digital-to-analog converters consisting of the transistor voltage switches and the resistor networks.

With the transistor, time-division multiplier, and high-gain low-drift d-c-amplifiers, the static accuracy of the square-root function generator in Fig. 7.1.19 is better than

±0.1 per cent of the full-scale output (±10 volts) for all values of X, from zero to full scale. If d-c amplifier 1 drifts, the square root of the drift voltage will appear at the output. The dynamic performance of this unit is essentially a function of that of the transistor time-division multiplier. The binary-increment multiplier of Fig. 7.1.11*a* may be used in a similar circuit.

Generation of $V_o = (X^2 + Y^2)^{1/2}$. A direct application and extension of the square-root function generator described above is the circuit illustrated in Fig. 7.1.20. The transistor time-division multiplier is connected to solve the equation

$$\frac{X^2}{V_o + Y} + 2Y - (V_o + Y) = 0$$

at the input of d-c amplifier 2. The quantity $2Y$ is added into amplifier 2 by conventional means. The output from amplifier 2, $-(V_o + Y)$, is fed back to the transistor switch in the feedback loop. This will make the output from the pulse-width modulator, which is the duty ratio of the pulses, proportional to $X/(V_o + Y)$. When this term is multiplied by X, the first term of the equation above is obtained. D-C amplifier 3, which subtracts Y from $V_o + Y$, may be part of another circuit.

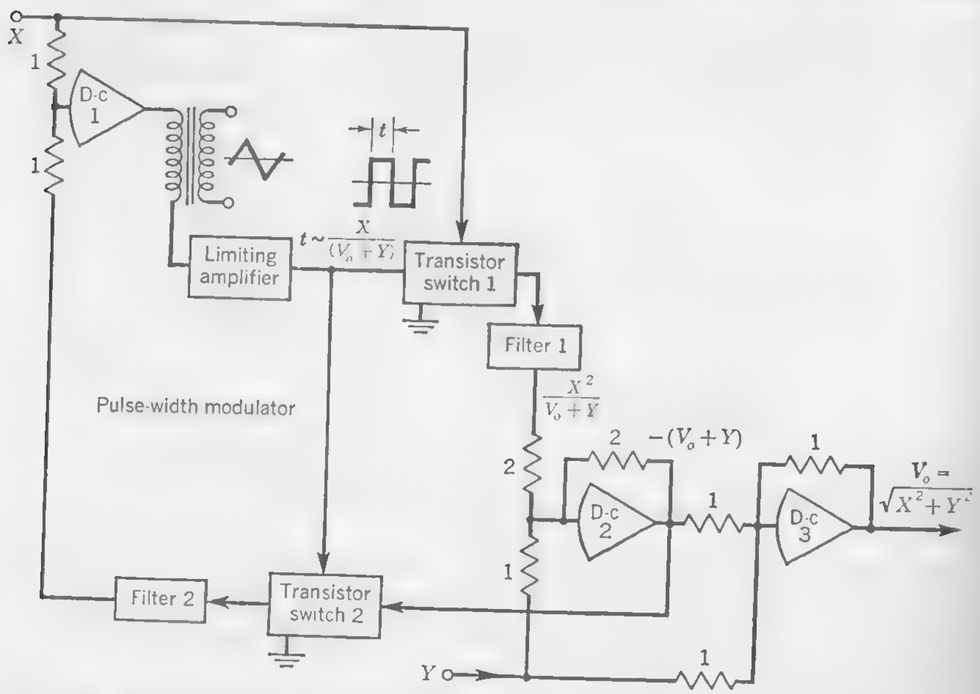


FIG. 7.1.20. Generation of $V_o = K\sqrt{X^2 + Y^2}$ with the transistor time-division multiplier.

The magnitude of $-(V_o + Y)$ must be smaller than the breakdown voltage of the switching transistors (25 volts) in switch 2. This limits the full-scale output to less than half this breakdown voltage. Two-quadrant operation is feasible if $+(V_o + Y)$ and $-(V_o + Y)$ are generated and fed back to the switch in the feedback loop. Four-quadrant operation is, in general, possible too, but requires several additional d-c amplifiers.

The static accuracy of this triangle-solving device is, in general, better than ±0.1 per cent of full-scale output. The dynamic performance is similar to that of the closed-loop time-division multiplier.

7.1.17. Electronic Rectangular-to-polar Coordinate Conversion Unit. The circuit of Fig. 7.1.21*a* accepts the input variable X and Y as d-c voltages and generates a d-c

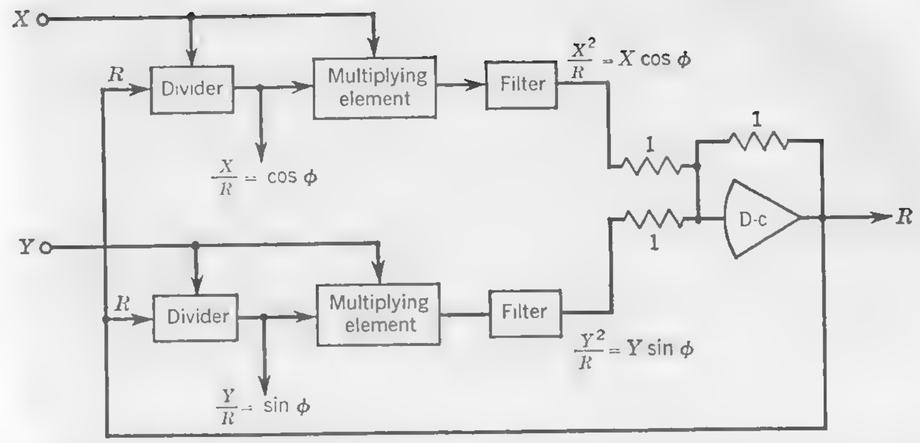


FIG. 7.1.21*a*. Simplified block diagram of an all-electronic rectangular-to-polar coordinate conversion unit.

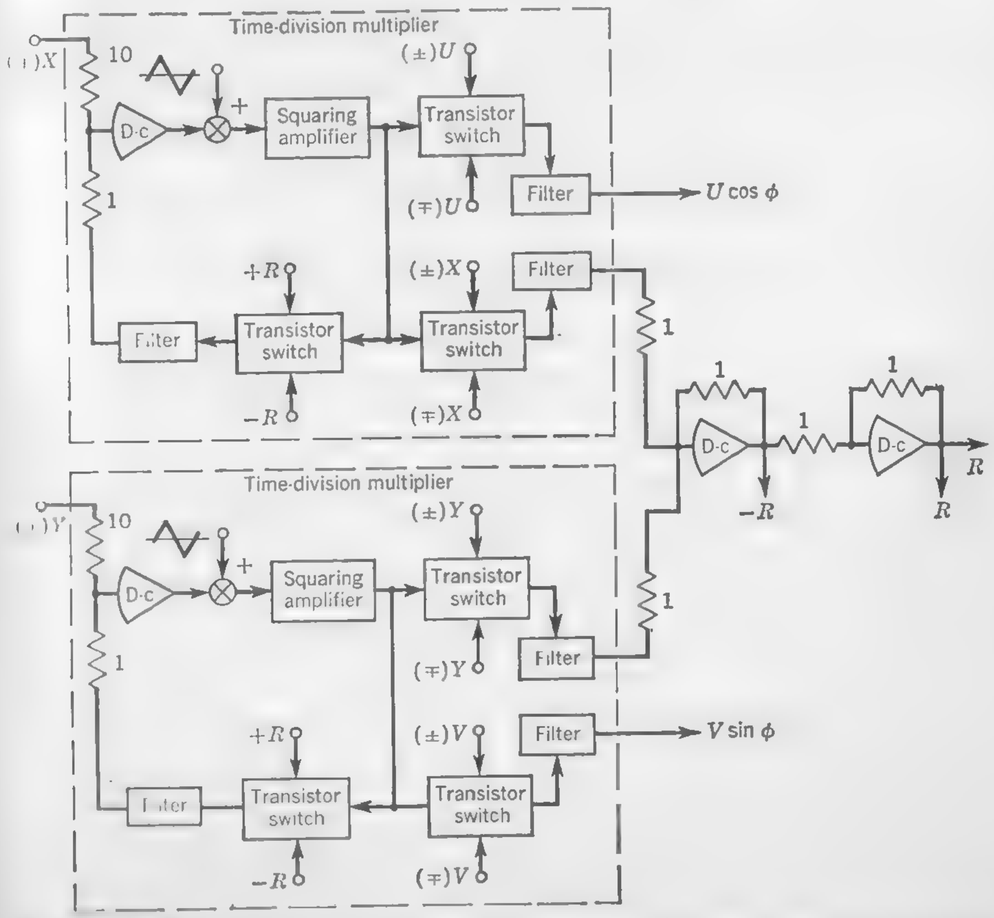


FIG. 7.1.21*b*. Block diagram of a four-quadrant rectangular-to-polar coordinate conversion unit using two transistor time-division multipliers.

voltage proportional to $R = (X^2 + Y^2)^{1/2}$. Instead of the angle $\phi = \arctan Y/X$, the circuit generates the sine and the cosine of this angle. This is a special advantage since, in most analog-computer applications, the angle must be converted back into the trigonometric functions. Another advantage is that the sine and cosine appear, for example, as the duty ratio of a pulse train when the transistor time-division multiplier is used, so that these outputs are easily multiplied by other variables with the aid of additional transistor voltage switches.

The simplified block diagram of Fig. 7.1.21*a* illustrates that the coordinate converter can be built with any two feedback multipliers capable of generating XY/Z. If the X variable and the R variable are connected to the inputs of a divider, the output from the divider is X/R, which is the cosine of the angle ϕ. Similarly, if Y and R are the inputs, the output is $Y/R = \sin \phi$. Multiplication of X/R by X and Y/R by Y gives X^2/R and Y^2/R, respectively. Addition of these two terms yields

$$\frac{X^2}{R} + \frac{Y^2}{R} = R = X \cos \phi + Y \sin \phi \tag{7.1.4}$$

Four-quadrant operation can be obtained most easily by providing $+R$ and $-R$ and feeding both back to the divider circuits. This is illustrated in Fig. 7.1.21*b*, where the transistor time-division multiplier is used. The divider circuits there consist of the pulse-width modulator and the multiplying elements of transistor voltage switches.

The static accuracy and the dynamic performance of this rectangular-to-polar coordinate conversion unit in Fig. 7.1.21*b* are similar to those of the time-division multiplier. A static accuracy of ± 0.1 per cent of full scale can be maintained over all four quadrants for both R, $k \cos \phi$, and $k \sin \phi$.

If an output voltage proportional to the angle between X and Y is required, the sine or cosine of ϕ can be converted into the angle by applying the arcsine generator described in Art. 7.1.14. The accuracy of the latter conversion is, however, limited to ± 0.5 per cent and the range of operation to ± 90 deg.

The rectangular-to-polar coordinate conversion unit can be extended to perform in a three-dimensional system simply by cascading similar circuits.

References

1. Korn, Granino A., and Korn, T. M.: *Electronic Analog Computers*, McGraw-Hill Book Company, Inc., New York, 1956.
2. Kerfoot, B. P.: Transistors in Current-analog Computing, *IRE Trans. PGEC*, June, 1956.
3. Stanton, J. W.: Transistorized A-C Amplifiers, *IRE Trans. PGCT*, March, 1956.
4. Schuster, D.: D-C Transistor Amplifiers, *Electronic Eng.*, February, 1958.
5. Hunter, L. P.: *Handbook of Semiconductor Electronics*, sec. B, McGraw-Hill Book Company, Inc., New York, 1956.
6. Matzeu, W. T., and J. R. Biard: Differential Amplifier Features D-C Stability, *Electronics*, **32** (Jan. 16, 1959).
7. Chaplin, G. B. B., and A. R. Owens: A Transistor High-gain Chopper Type-D.C Amplifier, *J. Brit. Inst. Elec. Eng.*, Nov. 21, 1957.
8. Shea, R. F.: *Transistor Circuit Engineering*, chap. 5, John Wiley & Sons, Inc., New York, 1957.
9. New Look at Modular Quads, Transistor Application Ideas, PB-26, *Transistron Electronic Corp.*, Wakefield, Mass.
10. Ebers, J. J., and J. L. Moll: Large Signal Behavior of Junction Transistors, *Proc. IRE*, December, 1954, pp. 1761–1772.
11. Bright, R. L.: Junction Transistors Used as Switches, *Communication and Electronics*, March, 1955, pp. 111–121.
12. Sheenan, W. E.: Some Practical Considerations Concerning the Limitations of Junction Transistors, *Raytheon Tech. Bull.*
13. Kruper, A. P.: Switching Transistors Used as Substitute for Mechanical Low Level Choppers, *Communication and Electronics*, March, 1955, pp. 141–144.
14. Moody, N. F.: A Silicon Junction Diode Modulator of 10^{-8} Ampere Sensitivity for Use in Junction Transistor DC Amplifiers, *Proc. Natl. Electronics Conf.*, **11**:441–451 (October, 1955).

15. Schmid, H.: A Transistorized Four-quadrant Time-division Multiplier with an Accuracy of 0.1 per cent, *IRE Trans. on Electronic Computers*, **EC-7** (March, 1958).
16. Bibby, R. J., and P. M. Thompson: A High-speed Analogue Multiplier with a Linearity of Better than $\pm \frac{1}{3}$%, paper presented at the Solid State Circuit Conference, Philadelphia, February, 1958.
17. Meyers, R. A., and H. B. Davis: Triangular Wave Analog Multiplier, *Electronics*, August, 1956, pp. 182–185.
18. Palevski, M.: A Solid State Analog to Digital Conversion Device, *IRE Natl. Convention Record*, March, 1958, pp. 232–235.
19. Susskind, A. K.: *Note on Analog to Digital Conversion Techniques*, Technology Press, MIT, Cambridge, Mass., 5-32, 1957.
20. Strutt, M. H.: Hall Effect on Semiconductor Compounds, *Brit. Electronic & Radio Engineer*, January, 1959, pp. 2–10.
21. Chasmar, R. P., and E. Cohen: An Electrical Multiplier Utilizing the Hall Effect in Indium Arsenide, *Electronic Eng.*, December, 1958, pp. 661–664.
22. Hilsum, C.: Multiplication by Semiconductors, *Electronic Eng.*, December, 1958, pp. 664–666.
23. Lofgren, L.: Analog Multiplier Based on the Hall Effect, *J. Appl. Phys.*, **29**:158 (February, 1958).
24. Oxeni, J.: Hall Effect Multipliers, *Nachrtech. Z.*, **11** (May, 1958).
25. Magnetoresistor, *Electronic Design*, Nov. 12, 1958.
26. Schmid, H.: A Transistorized, All-electronic Cosine/Sine Function Generator, *IRE ESCON Convention Record*, part 4, pp. 99–105, 1958.
27. Schwartz, J., and R. Solomonoff: Photoelectric Chopper for Guided Missiles, *Electronics*, **27**:162 (November, 1954).
28. Rote, W. A.: Magnetic Converter DC Amplifier, *Electronics*, **26**:170 (December, 1953).
29. Thomas, D. G. A.: A Simple Vibrating Condenser Electrometer, *Electronic Eng.*, **22**:395 (September, 1950).

PART 2: DESIGN OF A HIGH-QUALITY TRANSISTOR D-C AMPLIFIER*

By WALTER HOCHWALD *and* HAROLD L. EHLERS

REQUIREMENTS AND BASIC THEORY

7.2.1. Requirements. Many computer applications require d-c amplifiers with zero-level offsets less than 0.005 per cent of full output, input resistances of up to 1 megohm, and input sensitivities on the order of 100 $\mu\mu$a. Also, large dynamic output ranges on the order of ± 50 volts are generally desirable. It is, further, often mandatory to combine these stringent requirements with mechanical design features which provide the utmost in reliability and ruggedness, as well as minimization of weight, volume, and power consumption. This need, which exists primarily in airborne and portable applications and in installations involving a large number of amplifiers, has led to the development of d-c amplifiers which employ automatic balancing circuits (chopper amplifiers) and transistor circuitry throughout, inclusive of the low-level chopper or modulator.

7.2.2. Amplifier Parameters (see also Sec. 2.3). Referring to Fig. 7.2.1, the output voltage of a d-c amplifier with voltage feedback and n inputs is

$$E_o(s) = -Z_f(s)\left[\frac{E_1(s)}{Z_1(s)} + \frac{E_2(s)}{Z_2(s)} + \cdots + \frac{E_n(s)}{Z_n(s)}\right] + \frac{\dfrac{K(s)}{A(s)} Z_f(s)\left[\dfrac{E_1(s)}{Z_1(s)} + \dfrac{E_2(s)}{Z_2(s)} + \cdots + \dfrac{E_n(s)}{Z_n(s)}\right] - E_d K(s) - I_d Z_f(s)}{1 + \dfrac{K(s)}{A(s)}} \tag{7.2.1}$$

* The authors gratefully acknowledge the contributions of F. H. Gerhard of Autonetics, a Division of North American Aviation, Inc., Downey, Calif., to the material presented in this section. The circuit shown in Fig. 7.2.5 was developed by Mr. Gerhard at Autonetics.

where $K(s) = 1 + \dfrac{Z_f(s)}{Z_{in}(s)} + Z_f(s)\left[\dfrac{1}{Z_1(s)} + \dfrac{1}{Z_2(s)} + \cdots + \dfrac{1}{Z_n(s)}\right]$

$Z_{in}(s)$ = input impedance of amplifier
$-A(s)$ = gain of amplifier
E_d = d-c zero-level offset or drift referred to input of amplifier
I_d = stray input current
$E_1(s), E_2(s) \ldots E_n(s)$ = input voltages
Z_f = feedback impedance
$Z_1, Z_2, \ldots, Z_n$ = input impedances

Only the first term is desired; the next term is the error due to finite amplifier gain, followed by terms representing the effect of voltage offset and stray input current.

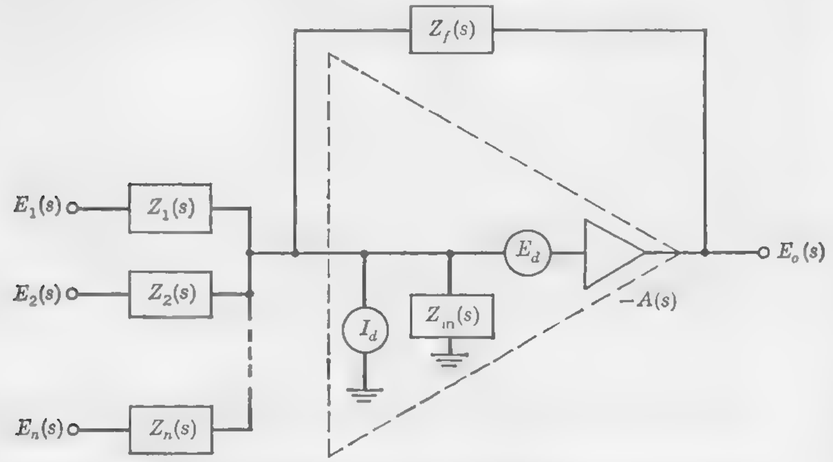


FIG. 7.2.1. D-C amplifier with over-all feedback and several inputs.

With 1-megohm input and feedback resistors, an accuracy of 0.005 per cent with respect to maximum output is obtained if the following parameter values can be assumed:

$Z_{in}(s)$ = 1 megohm
$A(s)$ = 5×10^6 at d-c
E_d = 3×10^{-4} volts
I_d = 10^{-9} amp
E_o(max) = ±45 volts

Equation (1) can be applied to any d-c amplifier configuration with or without over-all feedback. Omission of over-all feedback, for example, and consideration of one input voltage only, result in

$$E_o(s) = -\frac{E_1(s)A(s)Z_{in}(s)}{Z_1(s) + Z_{in}(s)} - E_dA(s) - \frac{I_dA(s)Z_1(s)Z_{in}(s)}{Z_1(s) + Z_{in}(s)} \qquad (7.2.2)$$

7.2.3. Amplifier Configurations (see also Sec. 2.3). If an automatic balancing circuit (chopper amplifier) is used, as in Figs. 7.2.2 to 7.2.4, the value for E_d in Eq. (1) or (2) is, essentially, divided by the d-c gain of the automatic balancing circuit (chopper amplifier) to yield the reduced value

$$E_d' = \frac{E_d}{1 + A_1} \qquad (7.2.3)$$

where E_d' = reduced d-c zero-level offset or drift referred to the input of the amplifier
E_d = uncompensated d-c zero-level offset or drift referred to the input of the amplifier
A_1 = d-c gain of the automatic balancing circuit (chopper amplifier)

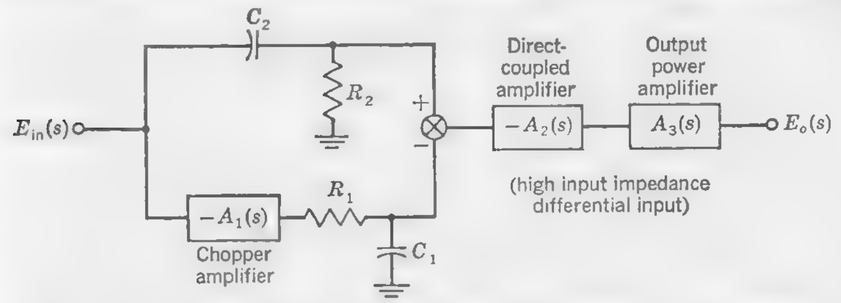


FIG. 7.2.2. Chopper-stabilized d-c amplifier configuration (conventional).

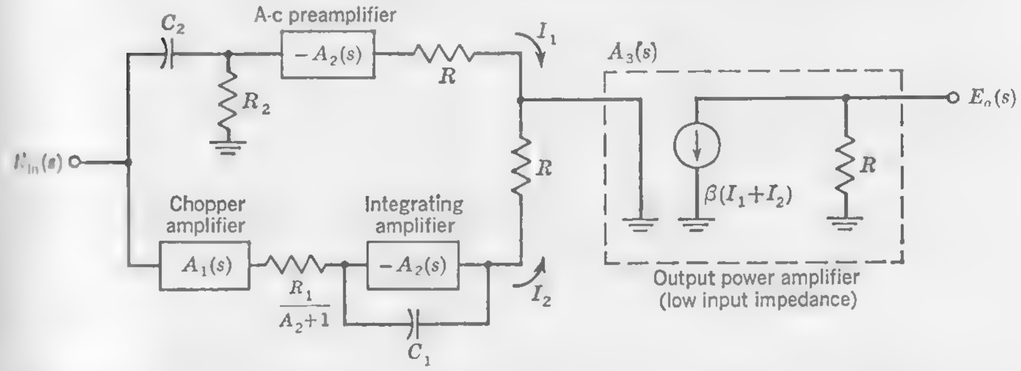


FIG. 7.2.3. Chopper-stabilized d-c amplifier configuration (modified to facilitate use of transistor circuitry).

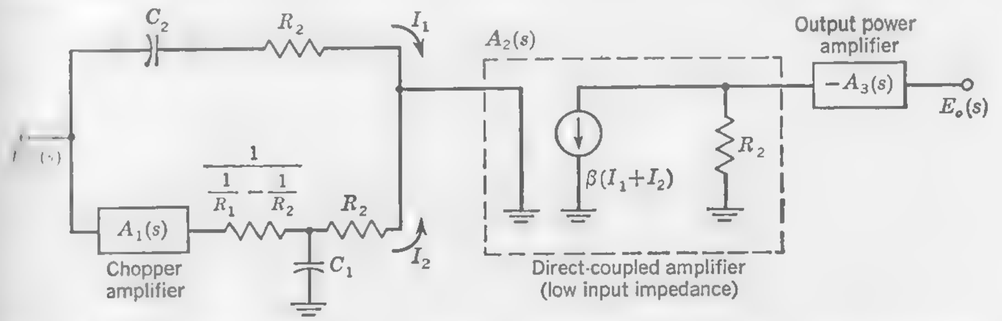


FIG. 7.2.4. Chopper-stabilized d-c amplifier configuration (modified to facilitate use of transistor circuitry).

If the effect of low-frequency variations of E_d on the amplifier output is of interest, it is necessary to take the coupling network (R_2,C_2) at the input of the direct-coupled amplifier and the filter (R_1,C_1) at the output of the chopper amplifier into account:

$$E_d'(s) = \frac{E_d(s)}{\dfrac{\tau_2 s}{\tau_2 s + 1} + A_1(s)\dfrac{1}{\tau_1 s + 1}} \qquad (7.2.4)$$

where $\tau_1 = R_1C_1$
$\tau_2 = R_2C_2$

In further detail, the block diagrams (Figs. 7.2.2, 7.2.3, and 7.2.4) represent three chopper-stabilized d-c amplifier configurations, which essentially result in identical transfer characteristics upon appropriate choice of amplifier gains and passive-network values.

$$A(s) = \frac{E_o(s)}{E_{in}(s)} = -A_2(s)A_3(s)\left[\frac{\tau_2 s}{\tau_2 s + 1} + A_1(s)\frac{1}{\tau_1 s + 1}\right] \qquad (7.2.5)$$

where $\tau_1 = R_1C_1$

$\tau_2 = R_2C_2$

$A_1(s)$, $A_2(s)$, $A_3(s)$ = gains of amplifier blocks

Figure 7.2.2 represents the conventional chopper-stabilized d-c amplifier configuration. Figures 7.2.3 and 7.2.4 are configurations which lend themselves more readily to use of transistor circuitry. For all three configurations, the amplifier sections are connected to provide two signal paths for the input signal $E_{in}(s)$. D-C and low-frequency components are preamplified in the chopper amplifier and low-pass filter (or integrating amplifier) combination, while the higher-frequency a-c components are amplified in the direct-coupled amplifier (or a-c preamplifier) and in the output power amplifier. The low- and high-frequency signals are combined either at the input of the direct-coupled amplifier (Figs. 7.2.2 and 7.2.4), or at the input of the output power amplifier in Fig. 7.2.3.

The effective preamplification of d-c input-signal components in the essentially "drift-free" chopper amplifier minimizes the effect of d-c drift in the succeeding direct-coupled circuitry, while provision of the alternate a-c signal path achieves amplification of high-frequency signals beyond the capabilities of the chopper amplifier-filter or integrating amplifier combination. The integrating amplifier in Fig. 7.2.3 provides a practical means for obtaining the long-time constant filter at the output of the chopper amplifier with relatively low resistor and capacitor values, because a time-constant multiplication by the gain of the integrating amplifier is effectively achieved. The filter thus formed, or the equivalent filters in Figs. 7.2.2 and 7.2.4, are required to remove chopper-frequency signals originating in the chopper amplifier.

The blocking capacitor C_2 is incorporated between the over-all amplifier input and the input of amplifier A_2 for all three versions. This prevents d-c flow from the input of amplifier A_2 back into the over-all amplifier input, and thereby eliminates this potential error input source.

Design of the amplifier blocks may now be discussed, primarily with reference to the configuration of Fig. 7.2.3, for which a complete circuit is shown in Fig. 7.2.5.

CIRCUIT DESIGN

7.2.4. Chopper Amplifier. The chopper-amplifier block is actually involved in all three configurations (Figs. 7.2.2, 7.2.3, and 7.2.4) and constitutes the most critical portion of the entire amplifier, because over-all accuracy is largely determined by the characteristics of the chopper. Figure 7.2.5 includes a chopper amplifier which employs a *p-n-p* microalloy silicon transistor chopper. Microalloy silicon transistor types are best suited for the chopper application because of their excellent high-frequency (switching) characteristics and the inherently small magnitude and stable nature of the internal sources of offset errors. A chopper operating frequency of around 350 cps is normally selected as a compromise: one must minimize high-frequency offset errors in the transistor chopper as well as the need for large coupling and by-pass capacitors. The transistor chopper Q_3 is operated in the inverted connection to minimize internally generated offset voltages. One further reduces offset effects by actuating the collector-base control junction with a half-wave-rectified square wave which alternately provides sufficient base current for the "closed-switch" state or zero voltage for the "open-switch" state. This rectified square wave is provided by the flux oscillator Q_1 and Q_2, and diode $CR1$, but could also be obtained from an external source capable of supplying several amplifiers, if desired. Operation with zero voltage across the control junction of the transistor chopper for the "open-switch" state is possible because the transistor "open-switch" resistance for *p-n-p* microalloy silicon transistors is quite high (generally greater than 50 megohms) for small d-c input signals to the chopper. Diodes $CR6$ and $CR7$ are used to clamp the input voltage to the chopper amplifier to less than 650 mv and thus avoid the need for back-biasing of the transistor chopper.

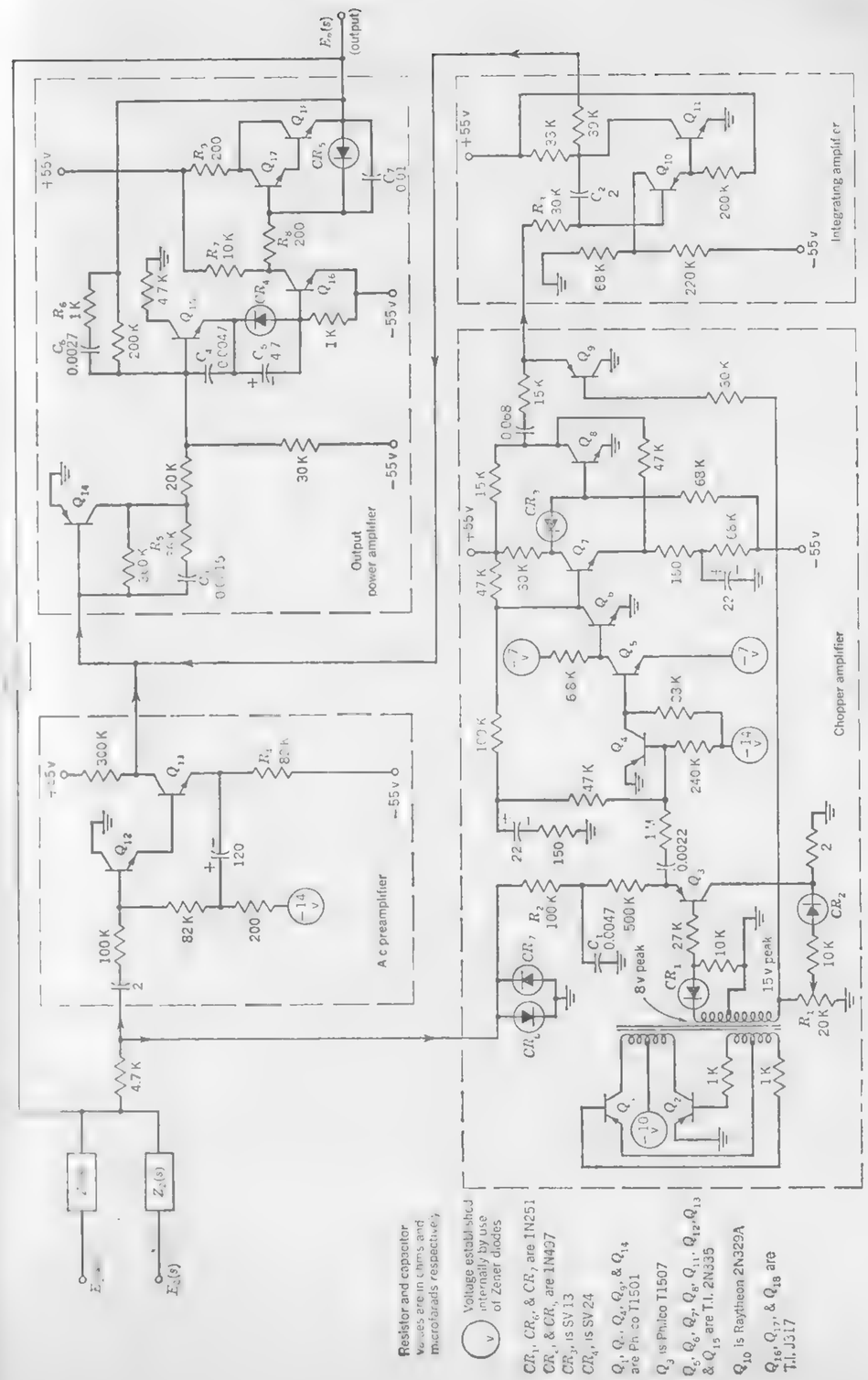

Output power amplifier
A c preamplifier
Integrating amplifier
Chopper amplifier
Resistor and capacitor values are in ohms and microfarads respectively
Voltage established internally by use of Zener diodes
CR1, CR6, & CR7 are 1N251
CR2, & CR5 are 1N497
CR3 is SV13
CR4 is SV24
Q1, Q2, Q4, Q9, & Q14 are Philco T1501
Q3 is Philco T1507
Q5, Q6, Q7, Q8, Q11, Q12, Q13 & Q15 are TI. 2N335
Q10 is Raytheon 2N329A
Q16, Q17, & Q18 are TI. J317

FIG. 7.2.5. D-C amplifier schematic.

Elimination of back-biasing eliminates temperature-dependent leakage-current errors normally observed in back-biased chopper circuits and is perhaps the most important single factor in the achievement of precision transistor low-level modulators. The transistor chopper circuit is further illustrated by Fig. 7.2.6; the following relationship exists:

$$V_d = -V_c - C_t V_t f R \tag{7.2.6}$$

where V_d = d-c voltage input required to reduce the a-c output of the chopper circuit to zero
V_c = equivalent voltage between base and emitter for saturated ("closed-switch") condition
C_t = capacitor representing approximately the lumped effects of junction capacity and minority carrier storage delays
V_t = peak value of voltage across C_t
R = input resistor to chopper circuit
f = switching frequency (chosen to permit complete discharge of C_t during each switching half cycle)

The stable nature of the components which comprise V_d makes it possible to include a zero control in the chopper circuit, which effectively eliminates modulator offset.

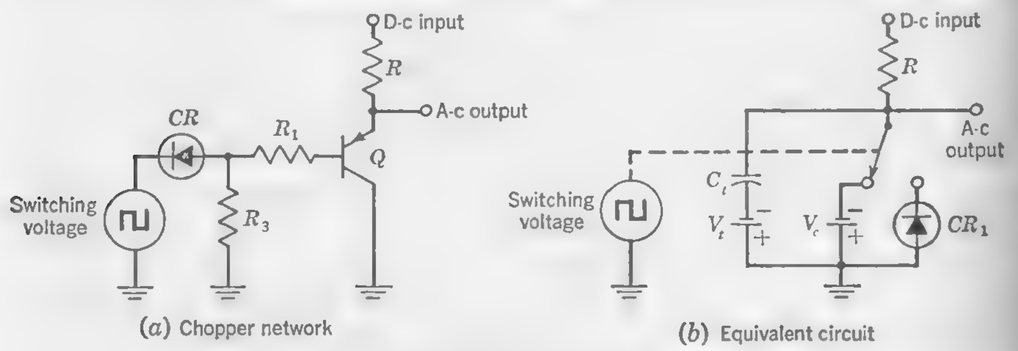


FIG. 7.2.6. Transistorized chopper circuit.

This is shown in Fig. 7.2.5, where the network consisting of R_1, $CR2$, and a 10-kilohm resistor inserts a variable square wave into the collector circuit of the transistor chopper. The zeroing voltage is applied only during the instant when the transistor chopper is in the conducting state; this again eliminates the possibility of errors due to leakage variations during the nonconducting state.

The chopper amplifier incorporates several other significant features:

1. The low-pass filter R_2C_1 is incorporated to prevent a-c components of the input signal from overdriving the high gain a-c amplifier.

2. The effects of switching transients or "spikes" due to junction capacity and carrier storage effects in the chopper transistor are reduced in the design of the a-c amplifier. Accurate clipping of the spike is provided at the output of the amplifier by means of saturation of Q_8 and control of the amplifier a-c input impedance and gain. This reduces the offset caused by the switching transient by approximately a factor of 3.

3. Bias stability in the a-c amplifier is provided through minimum use of large capacitors. Two feedback networks are incorporated—a low-pass network from the collector of Q_6 to the base of Q_4, and another from the collector of Q_8 to the emitter of Q_7. This arrangement also provides the stable input impedance and gain essential for setting the accurate clipping level of the switching transient mentioned in 2 above.

4. A one-transistor demodulator Q_9 is used to recover the d-c component of the signal. Blocking of large positive voltages is accomplished by providing a large switching voltage at the base of Q_9.

7.2.5. Integrating Amplifier. Figure 7.2.5 includes a schematic of the integrating amplifier. A time constant of approximately 12 sec is obtained with the aid of a two-transistor circuit and a feedback network comprising a 2-μf capacitor C_2 and a 30,000-ohm resistor R_3. The amplifier employs an n-p-n and p-n-p transistor circuit, which provides cancellation of the base-to-emitter voltage variations which are functions of ambient temperature. The base current of the input stage of the amplifier is minimized by employing a p-n-p silicon transistor which has high d-c beta at low collector currents. The minimization of base-to-emitter voltage variations, as well as the reduction of the input base current with resultant smaller-magnitude variations, is essential, because these variations effectively represent offset errors. These can be appreciable in spite of the fact that this portion of the circuit is preceded by the d-c gain of the chopper amplifier. The direct-coupled amplifier is bias-stabilized automatically, because it is in the d-c signal path of the over-all amplifier, which is normally stabilized by means of externally connected feedback networks.

7.2.6. A-C Preamplifier. The a-c preamplifier (Fig. 7.2.5) consists of an emitter follower Q_{12} that drives a common emitter amplifier Q_{13}. This arrangement provides an a-c input impedance of approximately 300,000 ohms. The large emitter resistor R_4 provides sufficient bias stability to reduce offset variations referred to the input of the over-all amplifier to a negligible value. n-p-n silicon transistors are employed because of their inherently good frequency-response characteristics, which permit extension of useful over-all amplifier gain to frequencies above 100 kc.

7.2.7. Output Power Amplifier. The essential feature of the output power amplifier is efficient provision of a large dynamic voltage range and sufficient current capacity. Utilization of a large dynamic voltage range effectively minimizes percentage errors due to offset and drift in most applications. The circuit included in Fig. 7.2.5 is also applicable to the configurations shown in Figs. 7.2.2 and 7.2.4.

With reference to Fig. 7.2.5, the output power amplifier consists of the composite emitter follower stage Q_{17} and Q_{18}, the common emitter amplifier Q_{16}, and a diode CR_5. For positive output voltages, the diode CR_5 is back-biased and may be considered as an open circuit. The circuit then consists of the composite emitter follower stages Q_{17} and Q_{18} driven by the common-emitter stage Q_{16}. For negative outputs, the diode is forward-biased to provide a low-resistance path from the collector of the common emitter stage Q_{16} to the load. The base-emitter junctions of Q_{17} and Q_{18} are back-biased, and they are therefore cut off. The circuit then functions as a common-emitter output stage. This arrangement results in an improvement in efficiency by a factor of over 100 compared with conventional single-transistor common-emitter or common-collector power-output stages.

Other salient features of the output power amplifier may be listed briefly:

1. Two power resistors R_8 and R_9 limit the output current and thereby provide protection of circuit components in the event of accidental shorting of the output.

2. Two additional gain stages Q_{14} and Q_{15} are included to perform voltage-level-shifting operations and achieve the required power gain.

3. Negative feedback is employed individually around Q_{14} and the remaining portion of the output power amplifier. This arrangement provides gain stability over the entire passband of the amplifier, which is essential for proper high-frequency cutoff control. Cutoff of the entire amplifier at 100 kc is accomplished by networks R_5C_3 and R_6C_6. These phase-control networks are designed to reduce the effects on frequency response due to beta cutoff and collector capacitance in the transistors of the a-c preamplifier and output power amplifier as well as parasitic capacitances in the layout. C_4, C_5, and C_7 stabilize the local feedback loop around Q_{15}, Q_{16}, Q_{17}, and Q_{18}.

4. Connection of Q_{14} as a common emitter amplifier with negative feedback provides the low input impedance required for efficient summation of the two signals from the integrating amplifier and the a-c preamplifier, respectively.

7.2.8. Specifications and Test Data. The complete d-c amplifier circuit is shown in Fig. 7.2.5. The amplifier circuit requires only two supply voltages, namely, +55 and −55 volts, although several convenient lower voltage levels are derived internally by means of Zener-diode voltage-divider networks, not shown in Fig. 7.2.5. The amplifier uses a total of 18 transistors. The d-c input resistance of the amplifier is approximately 1 megohm. The nominal d-c gain of the chopper amplifier is 1,300, the d-c gain of the integrating amplifier is 200, and the d-c gain of the output power

amplifier is 50; the result is a d-c open-loop gain of approximately 13 million. The voltage gain of the a-c preamplifier and power-output amplifier combined is approximately 3,000 at 100 cps, and 2,500 at 400 cps. The output voltage range is +45 to −45 volts direct current into a 1,500-ohm load.

Tests of the amplifier have shown that the offset remains less than 300 μv referred to the input, when the amplifier is operated over an ambient-temperature range of −55 to +85°C and less than ±50 μv over periods of several hundred hours under laboratory conditions. The transistor chopper Q_3 itself drifted less than ±10 μv during the test under laboratory conditions. Care in mechanical layout and guarding of the critical input circuit portions by grounded circuit lines generally results in leakage-current errors of less than 10^{-9} amp, with reference to the input, under conditions of high humidity.

Section 8

MISCELLANEOUS MECHANICAL AND ELECTRICAL ANALOG-COMPUTING SYSTEMS

By

WALTER W. SOROKA, *Professor of Mechanical Engineering, University of California, Berkeley, Calif.*

GRANINO A. KORN, *Professor of Electrical Engineering, University of Arizona, Tucson, Ariz.*

PAUL SAVET, *Staff Scientist, Arma Division, American Bosch Arma Corporation, Garden City, N. Y.*

CONTENTS

PART 1: MECHANICAL ANALOG COMPUTERS

Walter W. Soroka

Introduction

8.1.1. Mechanical Analog Computers 8–2
8.1.2. Error Sources 8–2

Representation of Basic Relations

8.1.3. Addition and Subtraction 8–2
8.1.4. Integration 8–2
8.1.5. Multiplication, Division, and Function Generation 8–5
8.1.6. Mechanical Differentiation 8–5
8.1.7. Scale Factors 8–5

Mechanical-differential-analyzer Techniques

8.1.8. Mechanical Differential Analyzers 8–11
8.1.9. Special Computer Setups 8–12

PART 2: A-C/ELECTROMECHANICAL ANALOG COMPUTERS

Granino A. Korn

Survey

8.2.1. A-C/Electromechanical Analog Computers. Representation of Variables 8–16
8.2.2. Computing Elements 8–16
8.2.3. Design of A-C/Electromechanical Computing Elements: References 8–16

A-C vs. D-C Analog Computers

8.2.4. Advantages of A-C Analog Computation 8–17
8.2.5. Disadvantages 8–17
8.2.6. Conclusions 8–17

PART 3: HEAT-TRANSFER COMPUTING ELEMENTS

Paul Savet

Introduction

8.3.1. Heat-transfer Computing Elements 8–18

Operating Principles

8.3.2. Heat-transfer Transducers 8–18
8.3.3. The Heat-transfer Multiplier-divider 8–19
8.3.4. Approximate Differentiation and Integration 8–20
8.3.5. Self-heating Transducers 8–20

PART 1: MECHANICAL ANALOG COMPUTERS

By WALTER W. SOROKA

INTRODUCTION

8.1.1. Mechanical Analog Computers. Mechanical analog computers consist of interconnected assemblages of sliders, bars, spur and bevel gears, noncircular gears, racks, screws, worms, tape-pulley combinations, variable-speed friction and pin-gear drives, and/or cams. Bar linkages limit the ranges of variables and the attainable accuracy but are preferred because they are easy to manufacture, are relatively low in cost, and give long trouble-free service.

Mechanical analog-computing elements represent variables by distances moved by specified points in the computer, or by angles turned by crank arms, gears, shafts, or screws. Both forms of representation are generally used in all but the simplest computers.

8.1.2. Error Sources. *Kinematical or "theoretical" errors* result from inherent approximations of the mechanization and may vary as the machine variables vary.

Fabrication errors result from manufacturing and assembly tolerances, and the necessary clearances to permit the proper operation of parts without interference or binding. As in other analog computers, the precision needed may have to exceed the required computer accuracy to prevent accumulation of errors.

Slip errors occur in friction drives (rollers, wheels, belts). They depend on operating conditions and usually increase with operating speed and loading.

REPRESENTATION OF BASIC RELATIONS

8.1.3. Addition and Subtraction. Figures 8.1.1 and 8.1.2 show a number of mechanical adder-subtractors. Such devices may be combined to yield sums or differences of three or more variables.

8.1.4. Integration. The mechanical integrators shown in Figs. 8.1.3 and 8.1.4 are friction-driven variable-speed devices. Although these integrators have no kinematical errors, they do have fabrication and slip errors.

To locate the plane of the wheel accurately, its rim should be sharp. This has been accomplished by crowning the rim with a radius of 0.002 in.,[25] or by using a 60-deg rim with its sharp edge ground to a 0.002-in. width,[27] or by crowning the rim with a radius of $\frac{1}{32}$ in.[26] To minimize wear and resistance to axial displacement of the wheel, hardened steel is commonly used for the wheel rim, and either polished hardened steel or polished plate glass for the turntable. This results in a friction drive of very low torque-transmission characteristics. In a practical case[25] the torque available at the wheel of a sharp-edged integrator wheel on a polished-glass disk was 0.0001 lb-ft. To drive attached equipment a torque amplification of the order of 10,000 was required. Torque amplification may be supplied by an all-mechanical capstan,[53] a polarized-light servo system,[37] or a capacitance-pickoff servo.[27] Tests at 1,000 wheel rpm on a disk-and-wheel integrator using a polarized-light servo system showed no appreciable slip with output torques up to 0.75 lb-ft.[37]

One usually determines the over-all effects of backlash and slip in an integrating system by generating a simple harmonic function and observing the change in amplitude of the function in successive cycles.[24] The growth in amplitude may be observed on counters on the lead screws of the integrators or, more conveniently, by plotting the cosine function against the sine function on an output table. For a perfect system this plot would be an exact circle. Amplitude-growth tests[37] on the integrators of a large-scale differential analyzer showed changes of the order of a small fraction of 1 per cent, particularly when a large gear-reduction ratio was used on the integrator lead-screw drive. The accuracy of the system could be further improved, if necessary,

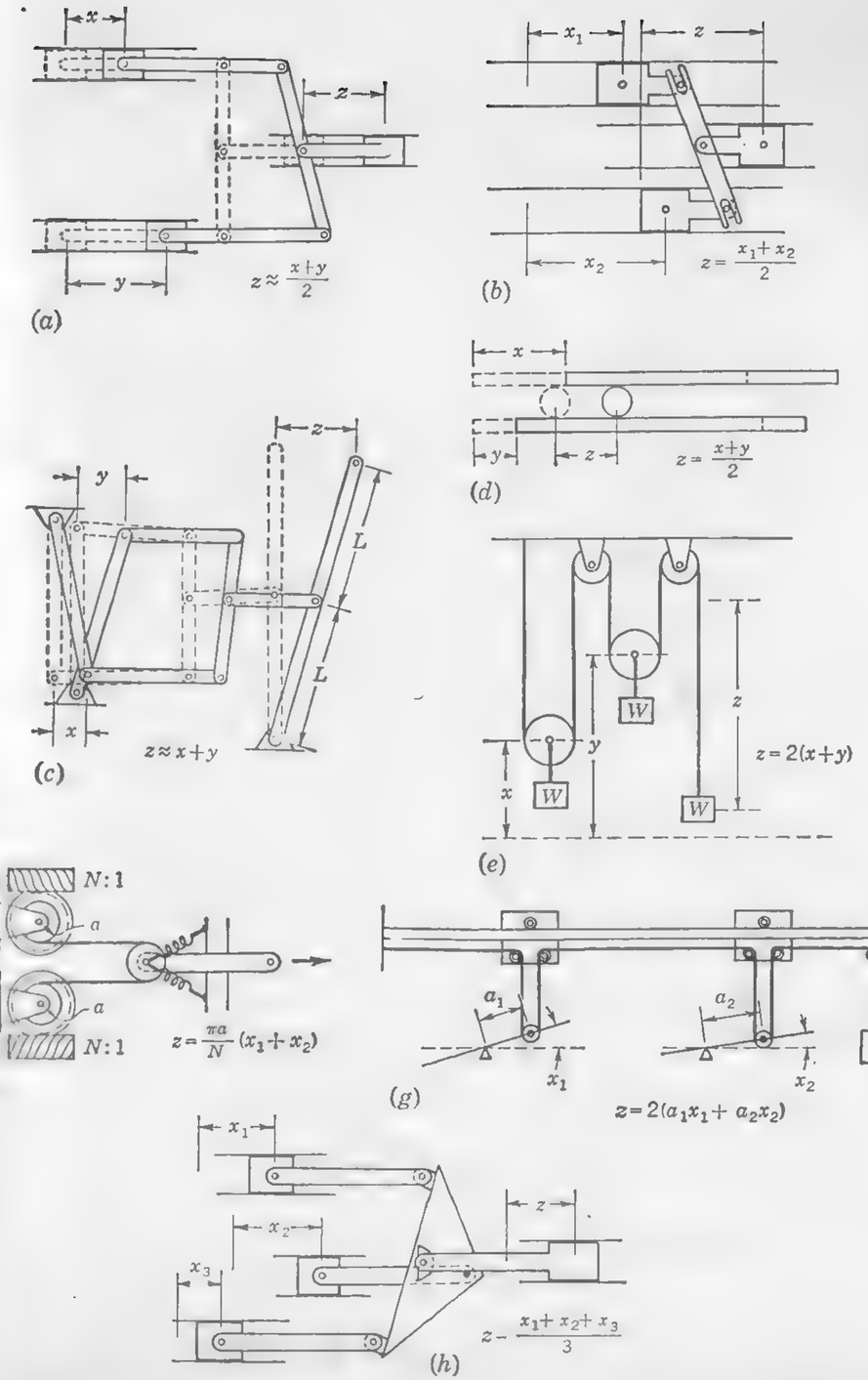


FIG. 8.1.1. Mechanical adder-subtractors. Items (*a*), (*c*), (*g*), and (*h*) have kinematical errors increasing with the amount of tilt in the linkage bars. The linkage adder (*c*) avoids the use of sliders and approximates addition of angular as well as linear displacements. For small angles of tilt in adder (*a*) the error is $(x - y)^4/128\ a^2L$, where $2a$ is the crossbar length (Refs. 1, 4, 7 to 11).

by the addition of antibacklash ("front-lash")[25,53] units, but this has rarely been found necessary.

Design criteria for *ball-and-disk integrators* (Fig. 8.1.3*b*) are substantially those for ball-bearing design with respect to surface finish, hardness, load, lubrication, and friction. Accurate alignment of the balls, and lapped surfaces, or roller guides, in the ball cage are necessary to minimize lateral thrust and friction in the cage. Ball size does not affect the integrator output turns. High pressure applied by loading springs often permits direct drives without torque amplifiers.

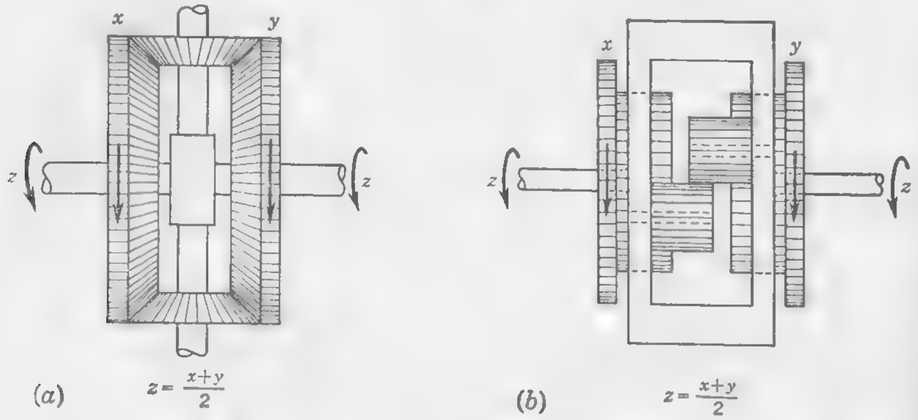


FIG. 8.1.2. Bevel-gear differential (*a*) and spur-gear differential (*b*). Combinations of differentials yield sums of more than two terms (Refs. 1, 4, 7 to 9, 12 to 17, 24, 26, 33 to 39, 53).

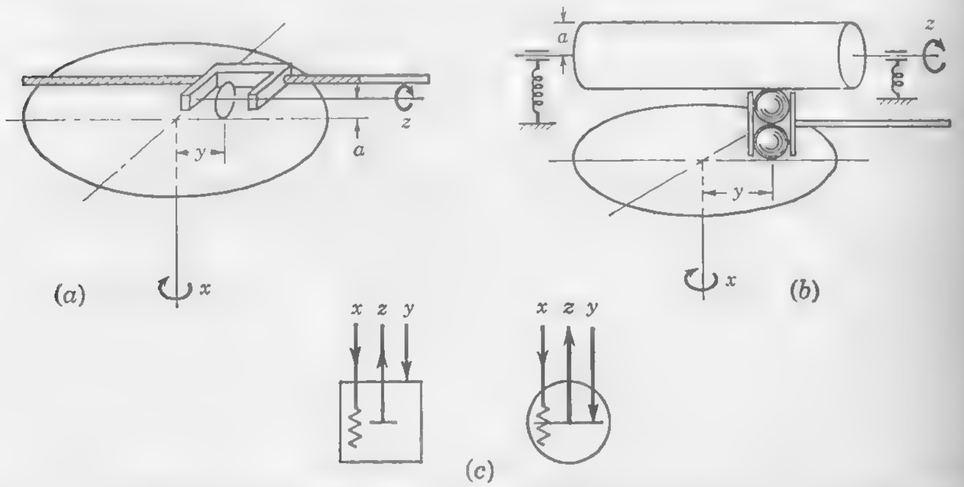


FIG. 8.1.3. Wheel-and-disk integrator (*a*), ball-and-disk integrator (*b*), and schematic symbols (*c*). The relation established is $dz = y\,dx$ (Refs. 2, 5, 8 to 10, 13, 14, 20, 25 to 39, 53).

The position of the point of contact between ball and disk is not determined as precisely as for the sharp-edged wheel. However, the ruggedness and torque-transmitting capacity of the ball-and-disk unit recommend its application to instrument, control systems, and computers for which extreme precision is not an overriding requirement. A sine-wave amplitude-growth test[41] on a small carefully constructed differential analyzer using ball-disk integrators without torque amplifiers showed an amplitude increase of approximately 0.2 per cent per cycle. This compares quite favorably with more elaborate disk-and-wheel differential analyzers incorporating torque amplifiers.

8.1.5. Multiplication, Division, and Function Generation. Figures 8.1.5 to 8.1.12 show a number of mechanical computing elements. The squaring cams of Fig. 8.1.6 have been used in quarter-square multipliers (Sec. 3.2). Where mechanical computing elements are nonreversible, one can often generate inverse functions implicitly by using a servomechanism in the manner of the high-gain amplifier in Fig. 5.1.6.

8.1.6. Mechanical Differentiation. In Fig. 8.1.13 spring deflections register the approximate time derivatives of angular displacements by measuring the torque

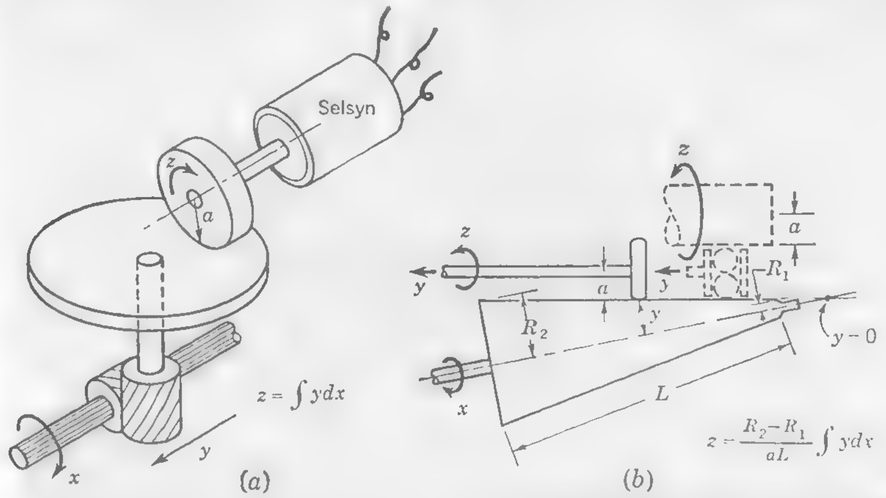


FIG. 8.1.4. Integrator employing a spherical-zone wheel, a sliding disk, and a selsyn pickoff (*a*); and cone integrator (*b*) (Refs. 20, 40, 43).

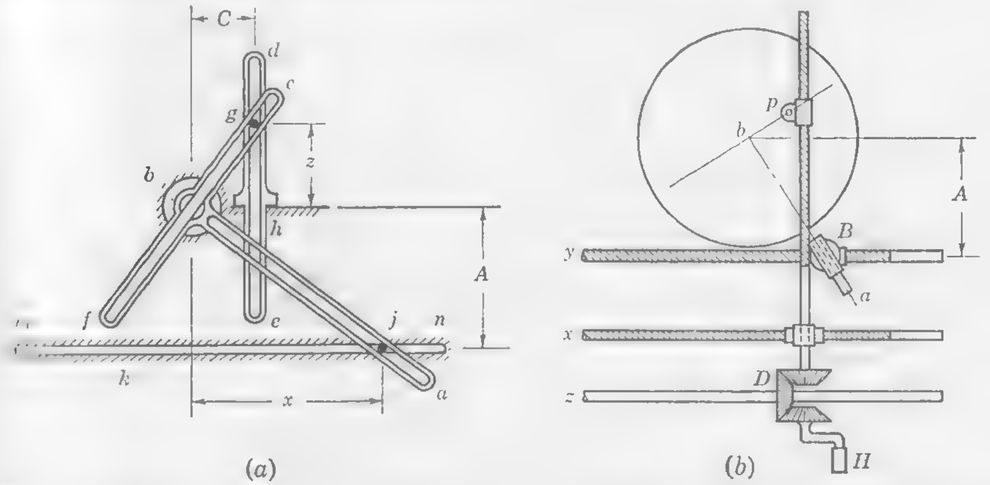


FIG. 8.1.5. Similar-triangle multipliers have no kinematical errors and operate in reverse for division. $z = xy/A$ (Refs. 2, 5, 8 to 10, 13, 14, 18, 46).

verted by a viscous-fluid drag cup (Fig. 8.1.13*a*), an eddy-current disk (Fig. 8.1.13*b*), and a rate gyro (Fig. 8.1.13*c*). Dashpots across the restraining springs serve to damp transient oscillations. Mechanical differentiation can also be obtained implicitly with the aid of a feedback loop containing a mechanical integrator in the manner of Art. [illegible] 8.1.3, 8.1.4, 8.1.7, 8.1.11, 8.1.37, 8.1.41

The *tangent of a traced curve* can be followed manually with the aid of a ruler or linkage bar bearing a small mirror normal to the curve.

8.1.7. Scale Factors. To keep linear and angular displacements within their ranges of travel, scale factors must be applied to the problem variables in the manner

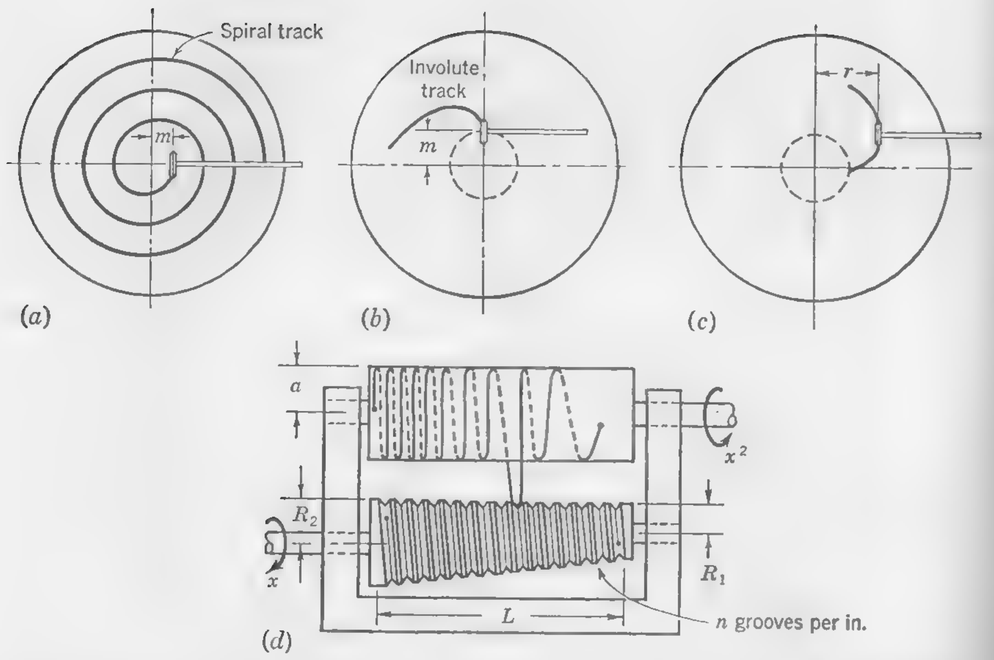


FIG. 8.1.6*a*, *b*, *c*. Toothed squaring cams. Input and output in (*a*) cannot go through zero; (*b*) and (*c*) show an involute track and offset wheel which permit the input to reach zero but no negative values. For a wheel of radius a one has $z = mx^2/2a$ (Refs. 8, 9, 18, 47).
FIG. 8.1.6*d*. Cone-squaring unit employing two sets of wires to permit rotation in either direction. Input cannot go through zero (Refs. 1, 9, 10, 14).

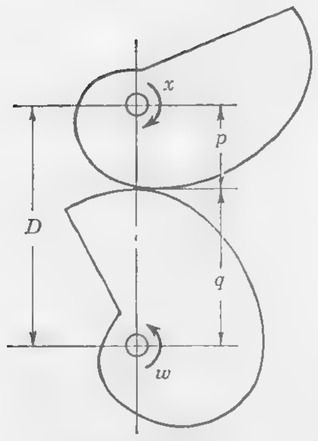


FIG. 8.1.7.

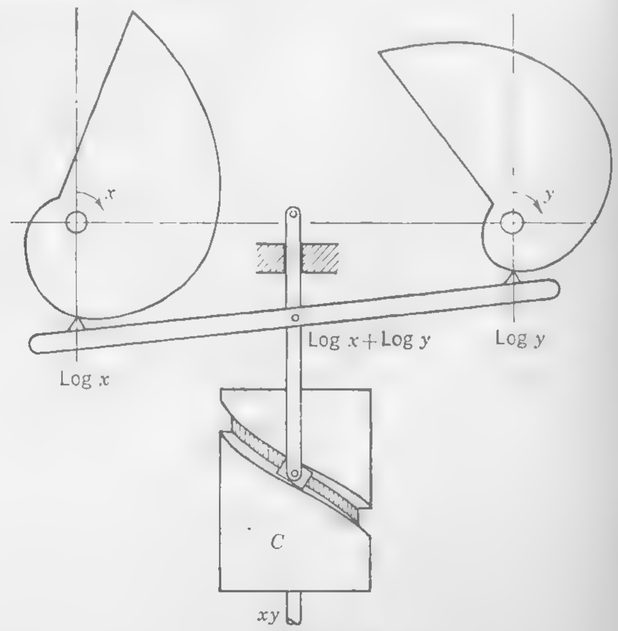
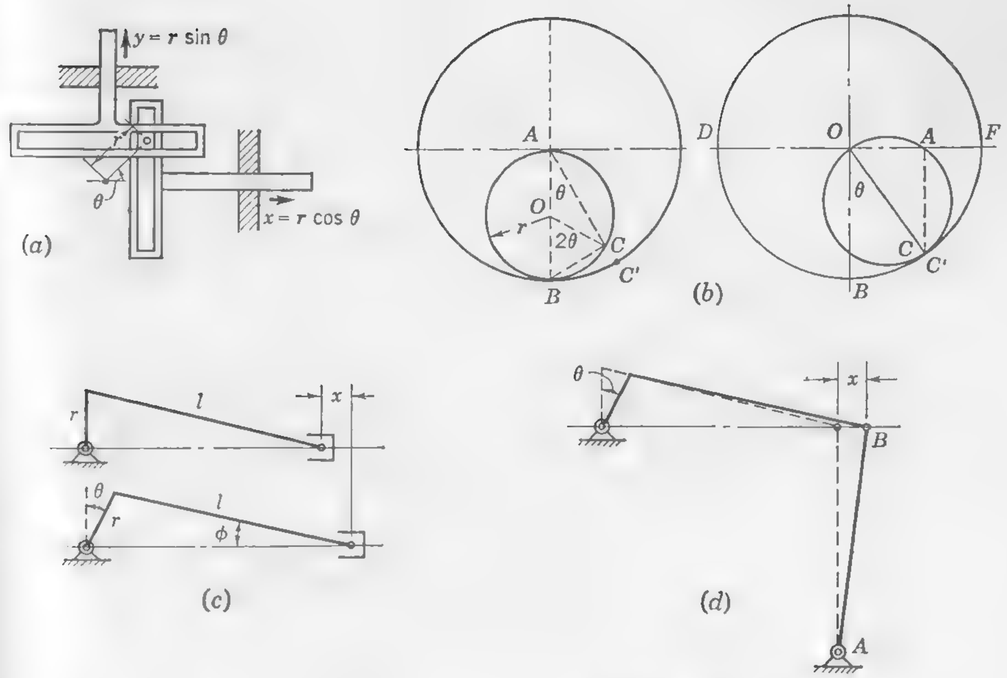


FIG. 8.1.8.

FIG. 8.1.7. Function cams will generate a function $w = w(x)$ (or its inverse) if

$$p = \frac{D}{1 + (dx/dw)} \qquad q = \frac{D}{1 + (dw/dx)}$$

Thus for $w = kx^2$ one has $dw/dx = 2kx$; and for $w = \log_e x$ one has $dw/dx = 1/x$. Cams may be toothed (Refs. 9, 19, 48 to 52).
FIG. 8.1.8. Logarithmic multiplier using three cams. Kinematical errors are due to tilt in rulder linkage (Refs. 2, 5, 9, 10, 18).



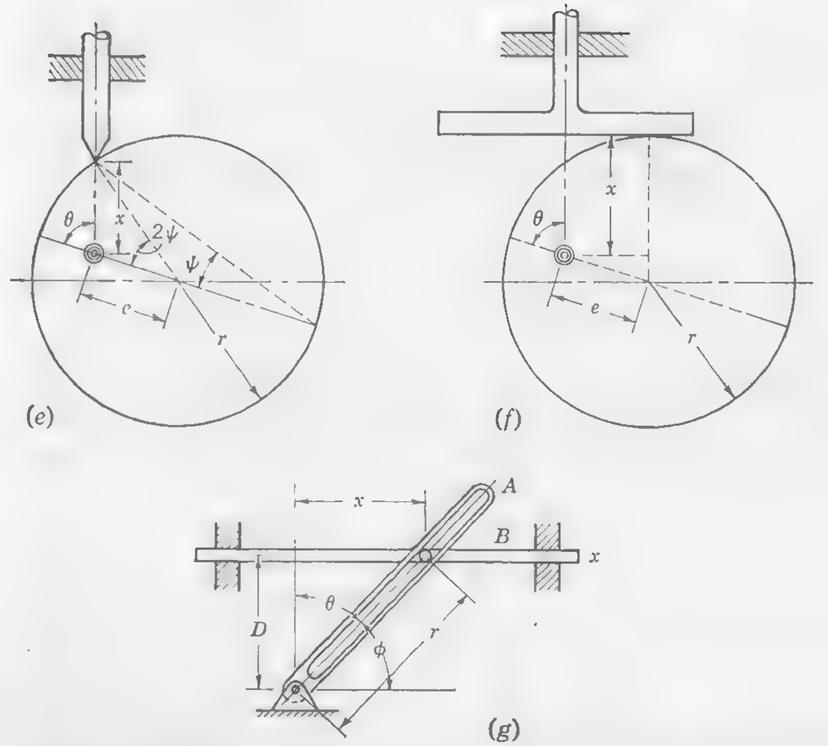


FIG. 8.1.9. Linkages and cams yielding displacements proportional to $\sin \vartheta$ and $\cos \vartheta$ (*a*) to (*f*) and $\tan \vartheta = \cot \phi$ (*g*). (*c*), (*d*), and (*e*) have kinematic errors, which are reduced in (*c*) by large values of l/r and in (*d*) and (*e*) by a shift of the reference line $\vartheta = 0$ (Refs. 4, 8 to 10, 18, 41).

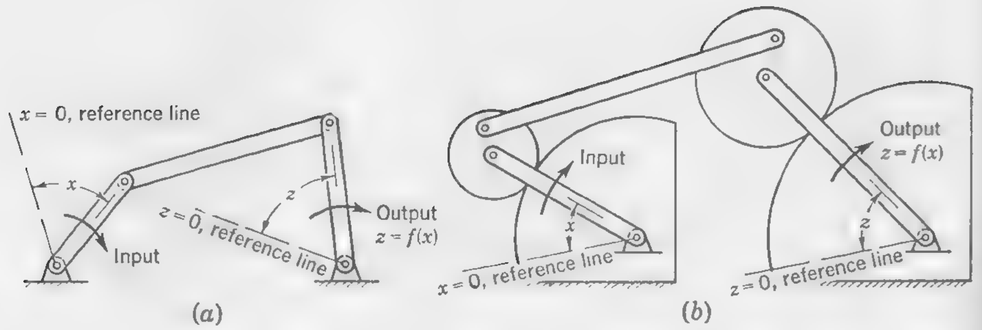


FIG. 8.1.10. Four-bar-linkage function generators. Modern synthesis methods suitable for digital-computer implementation permit design of such function generators with five "precision" points (points of zero kinematical error) and optimum spacing between precision points. Function generators for sin x, tan x, e^x, $x^{3/2}$, x^2, $x^{5/2}$, x^3, and others permit accuracies between 0.006 and 0.6 per cent of output travel. The eccentric linkages in Fig. 8.1.10*b* permit additional adjustments for reduction of kinematical errors (Refs. 8, 54 to 78, 93).

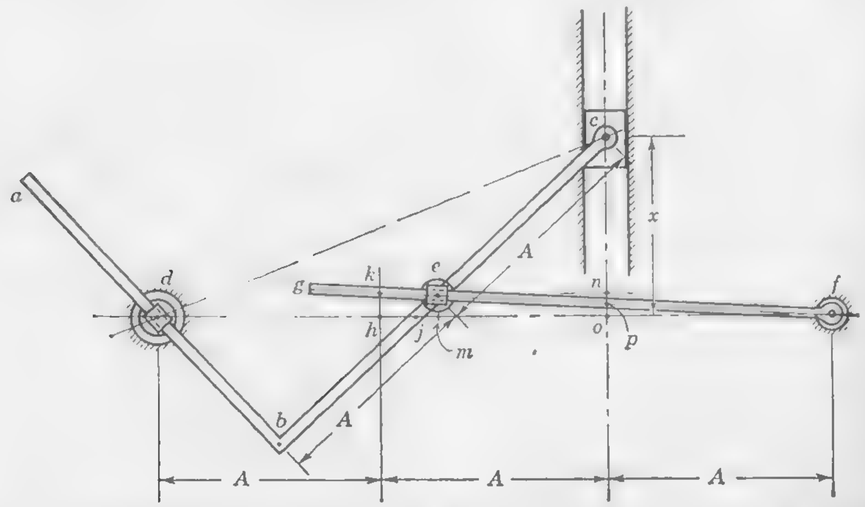


FIG. 8.1.11. Computing linkage based on the cissoid of Diocles. The rigid right-angle link *abc* has one end pivoted on a slider at *c*; the other leg of the right-angle link slides through a smooth collar pivoted at *d*. Another smooth collar is pivoted at the mid-point *e* of the leg *bc*; a bar *fg* pivoted at *f* slides in this collar. In the mechanism, $df = 3A$, while $bc = 2A$. Using $A = \frac{1}{2}$ as the unit of measure, the following relations are mechanized without kinematical errors:

$$hj = je = \frac{x^2}{2} \qquad en = \frac{1 - x^2}{2(1 + x^2)}$$

$$hk = x^3 \qquad fc = \frac{\sqrt{1 + x^6}}{1 + x^2}$$

$$fm = \frac{1}{1 + x^2} \qquad dc = \sqrt{1 + x^2}$$

$$me = \frac{x^3}{1 + x^2} \qquad jm = \frac{x^2 - x^4}{2(1 + x^2)}$$

$$bj = jO = \frac{1 - x^2}{2} \qquad np = \frac{x^3 - x^5}{2(1 + x^2)}$$

(Ref. 92).

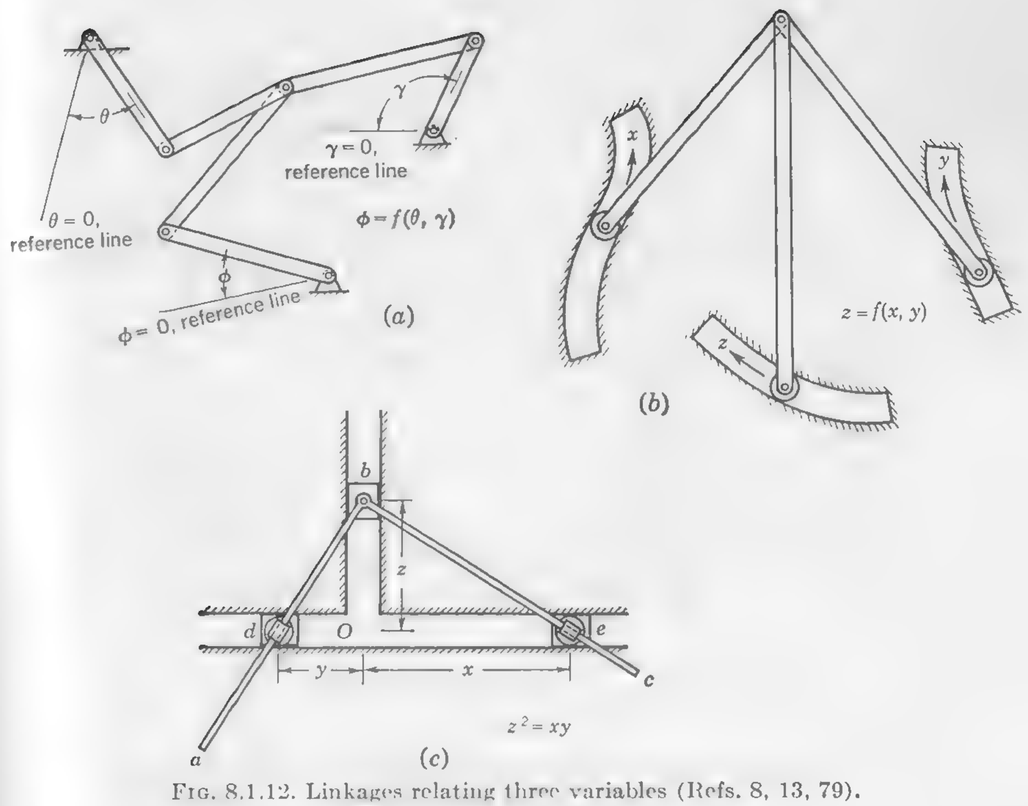


FIG. 8.1.12. Linkages relating three variables (Refs. 8, 13, 79).

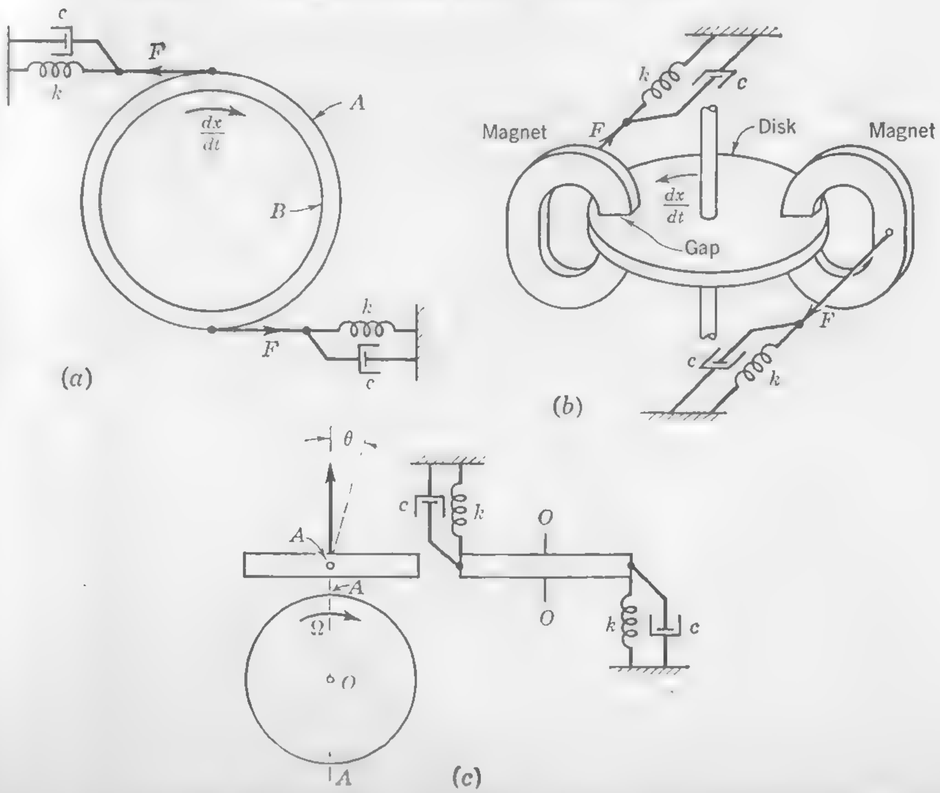
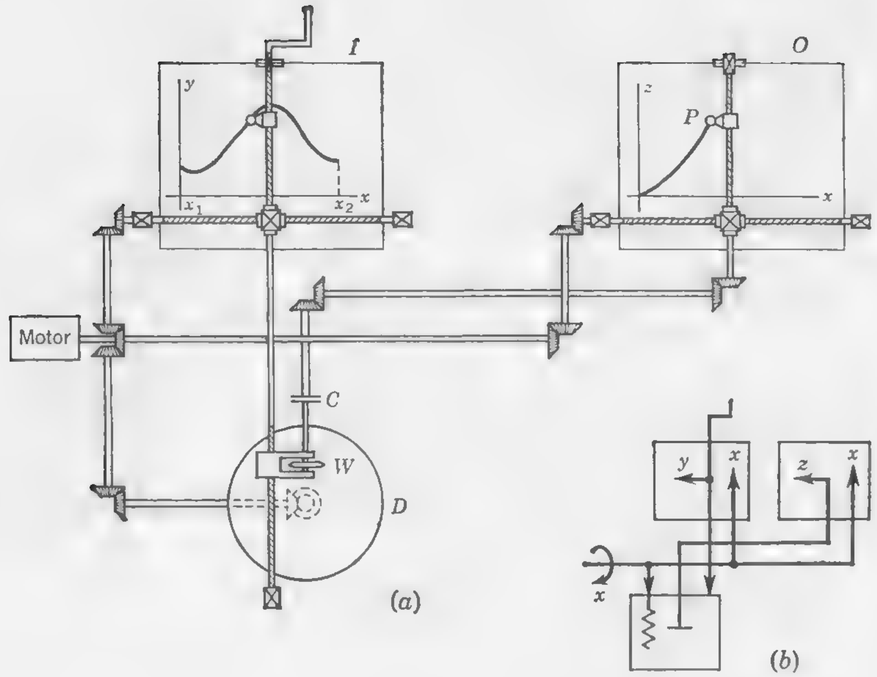


FIG. 8.1.13. Mechanical differentiation. The deflection of the restraining springs is approximately proportional to dx/dt or $d\theta/dt$ in the drag-cup differentiator (*a*), the eddy-current disk differentiator (*b*), and the rate gyro (*c*). Dashpots across the restraining springs prevent transient oscillations (Refs. 2, 5, 9, 14, 21, 37, 44).



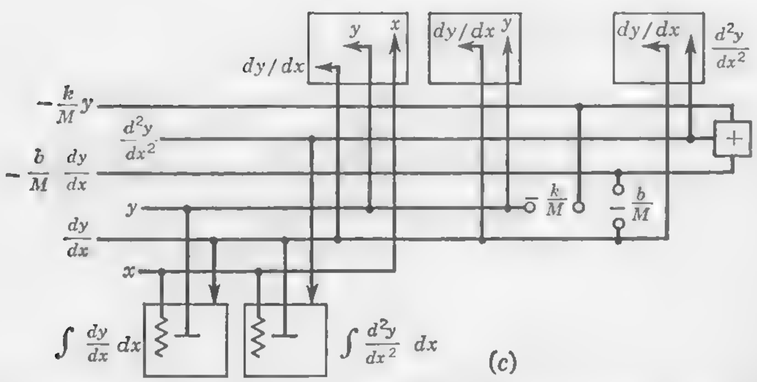


FIG. 8.1.14. Integration of a function on a mechanical differential analyzer using lead-screw-operated input and output tables (*a*), schematic representation (*b*), and setup for the solution of the differential equation $M\,d^2y/dx^2 + b\,dy/dx + ky = 0$ (*c*).

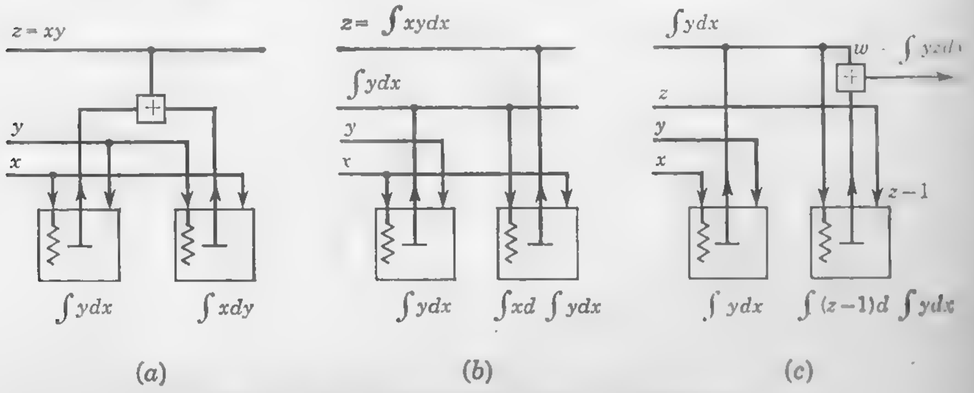
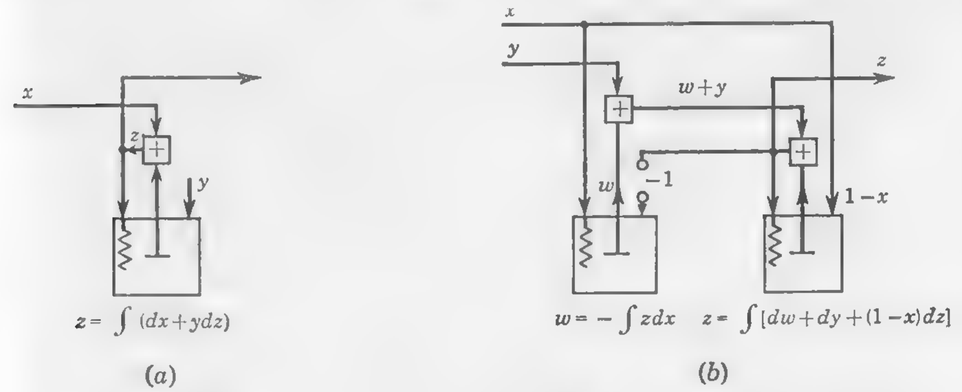


FIG. 8.1.15*a*. Multiplication with two integrators.
FIG. 8.1.15*b*, *c*. Integration of products.

of Sec. 5.1. Large-scale factors help to minimize the effect of fabrication errors; the same variable may have different scale factors in different parts of a computer.[27]

MECHANICAL-DIFFERENTIAL-ANALYZER TECHNIQUES

8.1.8. Mechanical Differential Analyzers. The pioneer differential analyzers using the techniques of Secs. 5.1 and 5.2 to solve ordinary differential equations as early as 1876 were mechanical devices employing disk-and-wheel integrators (Art. 8.1.4).[25 to 36]



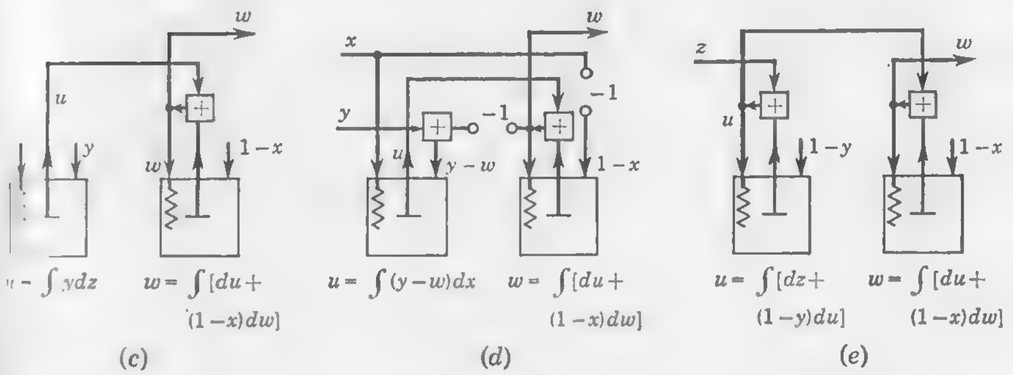


FIG. 8.1.16. "Regenerative" integrator connections.

(*a*) $z = \int \frac{dx}{1-y}$ (*b*) $z = \frac{y}{x}$ (*c*) $w = \int \frac{y}{x}\,dz$

(*d*) $w = \frac{1}{x}\int y\,dx$ (*e*) $w = \int \frac{dz}{xy}$

(*f*) $z = \log_e x$ (*g*) $z = \sqrt{2x}$ (*h*) $z = nx$

Denominators should not go through zero. In Fig. 8.1.16*h*, $z = nx$ is generated without a large gear train.

Multipurpose machines of this type have been rendered completely obsolete by the advent of electronic differential analyzers, which offer simpler programming, better analogy with signal-flow-diagrams, vastly increased computing speeds (runs measured in seconds rather than minutes and hours), and, with modern electronic multipliers, also better accuracy at lower cost. In high-accuracy applications, digital differential analyzers and multipurpose digital computers are preferred. In *special-purpose* analog computers mechanical-differential-analyzer techniques still are a powerful and

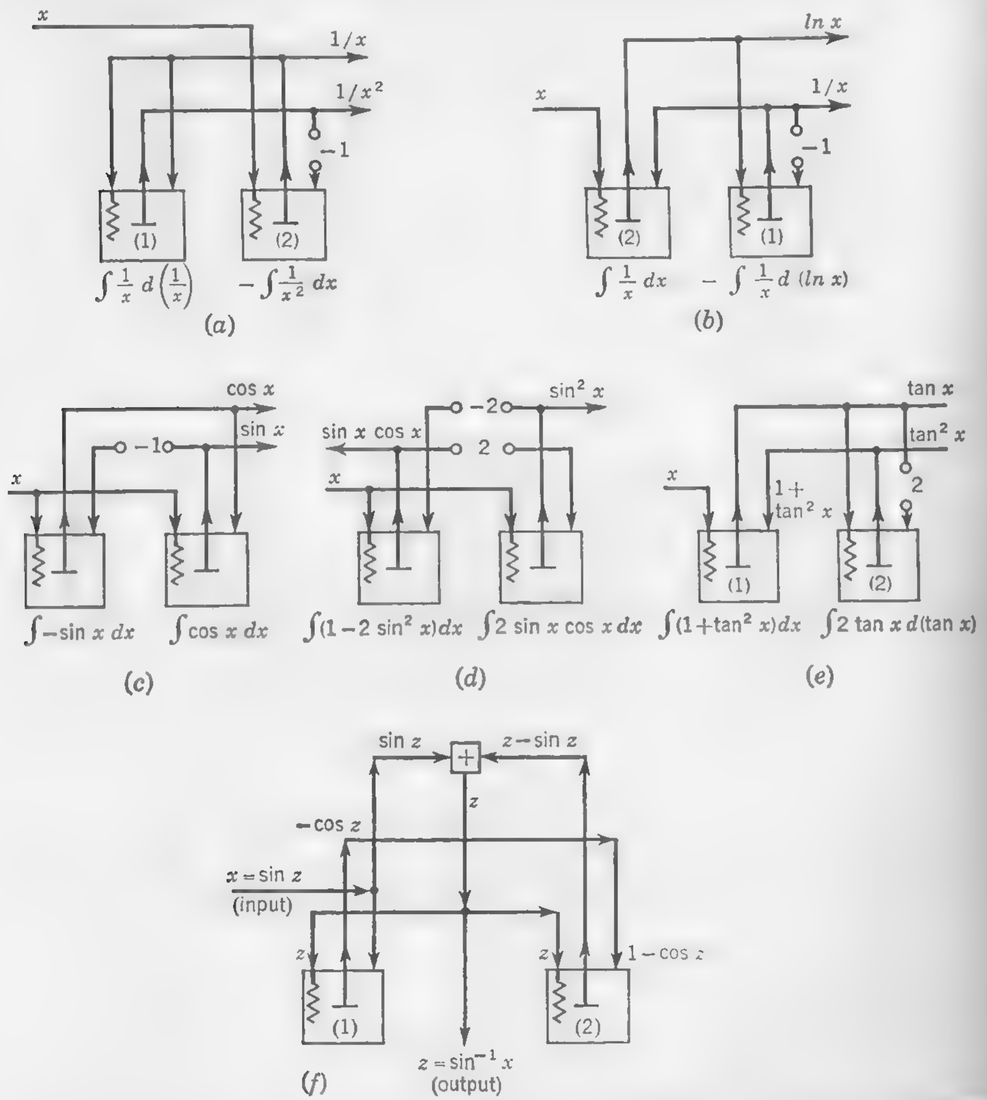


FIG. 8.1.17. Generation of various functions. Inverse hyperbolic functions may be generated by a computer setup essentially analogous to Fig. 8.1.17*f*.

accurate means of computation, particularly where shaft-rotation inputs and/or outputs are needed in any case. Four-digit accuracy is possible with favorable scale factors.

8.1.9. Special Computer Setups. Mechanical-differential-analyzer setups are slightly different from equivalent electronic-differential-analyzer setups because mechanical integrators integrate with respect to a shaft rotation (disk rotation) rather than with respect to time. One shaft rotation is usually chosen to represent the inde-

pendent variable, but integration with respect to other variables is possible. This permits multiplication by means of two integrators and generation of many functions of machine variables without special multiplying devices (Figs. 8.1.14 to 8.1.17). Many applications are described in the literature.[96 to 103] References 9 and 27 describe problem preparation, including a scale-factor treatment different from that described in Sec. 5.1.

References

1. Anonymous: Computing Mechanisms—I, *Product Eng.*, **27**(3):200–201 (March, 1956).
2. Anonymous: Computing Mechanisms—II, *Product Eng.*, **27**(4):180–181 (April, 1956).
3. Reid, R. R., and D. R. E. Stromback: Mechanical Computing Mechanisms—I, *Product Eng.*, **20**(8):131–135 (August, 1949).
4. Reid, R. R., and D. R. E. Stromback: Mechanical Computing Mechanisms—II, *Product Eng.*, **20**(9):119–123 (September, 1949).
5. Reid, R. R., and D. R. E. Stromback: Mechanical Computing Mechanisms—III, *Product Eng.*, **20**(10):126–130 (October, 1949).
6. Reid, R. R., and D. R. E. Stromback: Mechanical Computing Mechanisms—IV, *Product Eng.*, **20**(11):121–124 (November, 1949).
7. Kuhlenkamp, A.: Linkage Layouts by Mathematical Analysis, *Product Eng.*, **26**(8): 165–170 (August, 1955).
8. Svoboda, A.: *Computing Mechanisms and Linkages*, McGraw-Hill Book Company, Inc., New York, 1948.
9. Karplus, W. J., and W. W. Soroka: *Analog Methods in Computation and Simulation*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1959.
10. Dobrogurskii, S. O., and V. K. Titov: *Computing Devices* (in Russian), Government Publishing Agency for Defense Industries, Moscow, 1953.
11. Wilbur, J. B.: The Mechanical Solution of Simultaneous Equations, *J. Franklin Inst.*, **222**:715–724 (1936).
12. Davis, S. A.: Mechanical Components for Automatic Control, *Product Eng.*, **25**(9): 169–200 (September, 1954).
13. Billings, J. H.: Review of Fundamental Computer Mechanisms, *Machine Design*, **28**(3):213–216 (March, 1955).
14. Kobrinskii, N. E.: *Mathematical Machines of Continuous Action* (in Russian), Government Publishing Agency for Techno-Theoretical Literature, Moscow, 1954.
15. Giczewski, G. H.: Proportional Control System, *Machine Design*, **27**(12):216–220 (December, 1954).
16. Beam, A. S.: Beveloid Gearing, *Machine Design*, **27**(12):220–228 (December, 1954).
17. Fry, M.: Designing Computing Mechanisms, part I, Basic Elements, *Machine Design*, **17**(8):103–108 (August, 1945).
18. Fry, M.: Designing Computing Mechanisms, part II, Multiplying and Dividing, *Machine Design*, **17**(9):113–120 (September, 1945).
19. Fry, M.: Designing Computing Mechanisms, part III, Cam Mechanisms, *Machine Design*, **17**(10):123–128 (October, 1945).
20. Fry, M.: Designing Computing Mechanisms, part IV, Integration, *Machine Design*, **17**(11):141–145 (November, 1945).
21. Fry, M.: Designing Computing Mechanisms, part V, Differential Equations and Differentiation, *Machine Design*, **17**(12):123–126 (December, 1945).
22. Fry, M.: Designing Computing Mechanisms, part VI, Servomechanisms, *Machine Design*, **18**(1):115–118, 180 (January, 1946).
23. Fry, M.: Designing Computing Mechanisms, part VII, Stepping Followups, *Machine Design*, **18**(2):137–140 (February, 1946).
24. Crank, J.: *The Differential Analyser*, Longmans, Green & Co., Inc., New York, 1947.
25. Bush, V.: The Differential Analyzer, A New Machine for Solving Differential Equations, *J. Franklin Inst.*, **212**:447–488 (1931).
26. Kuehni, H. P., and H. A. Peterson: A New Differential Analyzer, *Trans. AIEE*, **63**:221–228 (1944) (Discussion, pp. 429–431).
27. Bush, V., and S. H. Caldwell: A New Type of Differential Analyzer, *J. Franklin Inst.*, **240**:255–326 (1945).
28. Bush, V., and H. L. Hazen: Integraph Solution of Differential Equations, *J. Franklin Inst.*, **204**:575–615 (1927).
29. Anonymous: The Differential Analyser, *Engineer*, **160**:56–58, 82–84 (1935).
30. Travis, I.: Differential Analyzer Eliminates Brain Fag, *Machine Design*, **7**(7):15–18 (July, 1935).

31. Hartree, D. R.: The Mechanical Integration of Differential Equations, *Math. Gaz.*, **22**:342–364 (1938).
32. Hartree, D. R., and A. K. Nuttall: The Differential Analyzer and Its Applications in Electrical Engineering, *J. Inst. Elec. Engrs.*, **83**:643–647 (1938).
33. Hartree, D. R., and A. Porter: The Construction of a Model Differential Analyzer, *Mem. Proc. Manchester Lit. & Phil. Soc.*, **79**:51–72 (1934–1935).
34. Lennard-Jones, J. E., M. V. Wilkies, and J. B. Bratt: The Design of a Small Differential Analyzer, *Proc. Cambridge Phil. Soc.*, **35**:485–493 (1939).
35. Massey, H. S. W., J. Wylie, R. A. Buckingham, and R. Sullivan: A Small-scale Differential Analyzer—Its Construction and Operation, *Proc. Roy. Irish Acad.*, (A), **45**:1–21 (1938–1939).
36. Beard, R. E.: The Construction of a Small-scale Differential Analyzer and Its Application to the Calculation of Actuarial Functions, *J. Inst. Actuaries*, **71**:193–227 (1941–1943).
37. Berry, T. M.: Polarized-light Servo System, *Trans. AIEE*, **63**:195–197 (1944).
38. Walther, A., and H. J. Dryer: Communications on Mechanical and Electrical Analyzers, *Proc. Inst. Mech. Engrs.*, **159**:74–76 (1948) (War Emergency Issue 38).
39. Janssen, E., and D. Lebell: Applications of a Mechanical Differential Analyzer to Electrical Engineering, *Elec. Eng.*, **70**:432–435 (1951).
40. Sorensen, E. G.: Construction and Maintenance Report on the UCRL Synchro Driven Differential Analyzer, *Univ. Calif. Radiation Lab. Rept.* UCRL-1717, February, 1952.
41. Charp, S.: A New Fourier Coefficient Harmonic Analyzer, *Trans. AIEE*, **68**:644–651 (1949).
42. Meyerott, R. E., and G. Breit: A Small Differential Analyzer with Ball Carriage Integrators and Selsyn Coupling, *Rev. Sci. Instr.*, **20**:874–876 (1949).
43. Tarpley, J. F., Jr.: A New Integrating Machine, *Military Engr.*, **32**:39–43 (1940).
44. Amble, O.: On a Principle of Connexion for Bush Integrators, *J. Sci. Instr.*, **23**:284–287 (1946).
45. Michel, J. G. L.: Extensions in Differential Analyzer Techniques, *J. Sci. Instr.*, **25**:357–361 (1948).
46. Cook, A. C., L. K. Kirchmayer, and C. N. Weygandt: Special Devices Aid Differential Analyzer Solutions of Complex Problems, *Trans. AIEE*, **69**:1365–1370 (1950).
47. Maxson, W. L., and P. J. McLaren: Multiplying Machine, U.S. Patent 2,194,477, Mar. 26, 1940.
48. Yavne, R. O.: High Accuracy Contour Cams, *Product Eng.*, **19**(8):134–136 (August, 1948).
49. Temperley, C.: Intermeshing Non-circular Algebraic Gears, *Engineering*, **165**:49–52 (1948).
50. Lockenvitz, A. E., J. B. Oliphant, W. C. Wilde, and J. M. Young: Noncircular Cams and Gears, *Machine Design*, **24**(5):141–145 (1952).
51. Hannula, F. W.: Designing Noncircular Surfaces, *Machine Design*, **23**(7):111–114, 190, 192 (July, 1951).
52. Peyrebrune, H. E.: Application and Design of Noncircular Gears, *Machine Design*, **26**(12):185–193 (December, 1953).
53. Rose, H. E.: The Mechanical Differential Analyser: Its Principles, Development, and Applications, *Proc. Inst. Mech. Engrs.*, **159**:46–54 (1948) (War Emergency Issue 38).
54. Freudenstein, F.: Four-bar Function Generators, *Machine Design*, **30**(24):119–123 (November 27, 1958).
55. Wolford, J. C., and J. C. Nicklas: Linkage Design Techniques, *Product Eng.*, **25**(12): 133–138 (December, 1954).
56. Hartenberg, R. S., and J. Denavit: Kinematic Synthesis, *Machine Design*, **28**(18): 101–105 (September 6, 1956).
57. Keator, F. W., and F. R. E. Crossley: 3-D Mechanisms, *Machine Design*, **27**(9):204–209 (September, 1955).
58. Dunk, A. C.: Designing Four-bar Linkages, *Machine Design*, **27**(2):195–199 (February; 1955).
59. Hall, A. S., and A. C. Dunk: Designing Four Bar Linkages to Substitute for Gears, *Machine Design*, **25**(4):233–236 (April, 1953).
60. Kraus, R. A., and K. S. Rangasami: Graphical Linkage Analysis, *Machine Design*, **25**(6):177–182 (June, 1953).
61. Nickson, P. T.: Linkage Design, *Machine Design*, **25**(7):137–144 (July, 1953).
62. Pike, E. W., T. R. Silverberg, and P. T. Nickson: Linkage Layout, *Machine Design*, **23**(11):105–110, 124–126 (November, 1951).

63. Nickson, P. T.: A Simplified Approach to Linkage Design, *Machine Design*, **25**(12): 196–209 (December, 1953).
64. Boehm, J.: Four-bar Linkages, *Machine Design*, **24**(8):118–122 (August, 1952).
65. Kaplan, J., and B. Pollick: Kinematic Analysis, *Machine Design*, **26**(1):153–160 (January, 1954).
66. Wundheiler, A. W.: Sound Formulas for Bar Linkages, *Proc. 1st Natl. Cong. Appl. Mechanics* (*U.S.*), pp. 19–22 (1952).
67. Hall, A. S.: Mechanism Proportions, *Machine Design*, **20**(2):111–115 (February, 1948).
68. Goodman, T. P.: Linkages vs. Cams, *Machine Design*, **30**(17):102–109 (August 21, 1958).
69. Hain, K.: Drag-link Mechanisms, *Machine Design*, **30**(13):104–113 (June 26, 1958).
70. Hall, A. S., Jr.: Alternate Four-bar Mechanisms, *Machine Design*, **30**(9):133–135 (May 1, 1958).
71. Pollitt, E. P.: Slider-crank Linkages, *Machine Design*, **30**(10):136–142 (May 51, 1958).
72. Hartenberg, R. S.: Complex Numbers and Four Bar Linkages, *Machine Design*, **30**(6):157–163 (March 20, 1958).
73. Martin, G. H.: Four-bar Linkages, *Machine Design*, **30**(8):146–149 (April 17, 1958).
74. Wolford, J. C.: Four-bar Linkages as Function Generators, *Product Eng.*, **26**(10): 166–171 (October, 1955).
75. Harding, B. L.: Four Bar Linkages, *Machine Design*, **29**(24):136–138 (November 28, 1957).
76. Freudenstein, F., and G. N. Sandor: Synthesis of Path-generating Mechanisms by Means of a Programmed Digital Computer, *J. Eng. for Ind.*, **81**, B:159–168 (1959).
77. Hain, K., and G. Marx: How to Replace Gears by Mechanisms (Linkages), *J. Eng. for Ind.*, **81**, B:126–130 (1959).
78. Freudenstein, F.: Structural Error Analysis in Plane Kinematic Synthesis, *J. Eng. for Ind.*, **81**, B:15–22 (1959).
79. Allen, C. W.: The Design of Linkages to Generate Functions of Two Variables, *J. Eng. for Ind.*, **81**, B:23–29 (1959).
80. Kloomok, M., and R. V. Muffley: Polynomial Cam Curves, *Product Eng.*, **28**(3):196–202 (March, 1957).
81. Maker, P.: Cam Design and Manufacture, *Machine Design*, **26**(12):188–190 (December, 1954).
82. Stoddart, D. A.: Polydyne Cam Design—II, *Machine Design*, **25**(2):146–154 (February, 1953).
83. Griffin, R. F.: Designing Cams with Aid of Computers, *Machine Design*, **25**(12):209–215 (December, 1953).
84. Rothbart, H. A.: Equivalent Mechanisms for Cams, *Machine Design*, **30**(6):175–180 (March 20, 1958).
85. Johnson, R. C.: Method of Finite Differences in Cam Design, *Machine Design*, **29**(23):159–161 (November 14, 1957).
86. Rothbart, H. A.: Basic Cam Systems, *Machine Design*, **28**(11):133–136 (May 31, 1956).
87. Johnson, R. C.: Cam Mechanisms, *Machine Design*, **28**(2):85–89 (January 26, 1956).
88. Weber, T.: Cam Development by Evolute Analysis, *Machine Design*, **28**(3):117–118 (February 9, 1956).
89. Cram, W. D.: Cam Design, *Machine Design*, **28**(22):93–103 (November 1, 1956).
90. Kloomok, M., and R. V. Muffley: Design of Cam Followers, *Product Eng.*, **27**(9): 197–201 (September, 1956).
91. Rothbart, H. A.: Basic Factors in Cam Design, *Machine Design*, **28**(21):107–113 (October 18, 1956).
92. Packer, L. S.: An Exact Computing Linkage, *Machine Design*, **25**(9):171–175 (September, 1953).
93. Hrones, J. A., and G. L. Nelson: *Analysis of the Four Bar Linkage: Its Application ot the Synthesis of Mechanisms*, John Wiley & Sons, Inc., New York, 1951.
94. Blackett, P. M. S., and F. C. Williams: An Automatic Curve Follower for Use with the Differential Analyzer, *Proc. Cambridge Phil. Soc.*, **35**:494–505 (1939).
95. Sauer, R., and H. Posch: Integriermaschine für gewohnliche Differential-gleichungen, *Z. Ver. deut. Ingr.*, **87**:221–224 (1943).
96. Hurty, W. C.: A Study of the Response of an Airplane Landing Gear Using the Differential Analyzer, *J. Aeronaut. Sci.*, **17**:756–764 (1950).
97. Blewett, J. P.: Tracing Electron Trajectories Using the Differential Analyzer, *Proc. IRE*, **36**:69–73 (1948).

98. Callendar, A., D. R. Hartree, and A. Porter: Time-lag in a Control System, *Trans. Roy. Soc.* (*London*), A, **235**:415–444 (1935–1936).
99. Concordia, C.: Network- and Differential-analyzer Solution of Torsional Oscillation Problems Involving Nonlinear Springs, *J. Appl. Mechanics*, **12**:43–47 (1945).
100. Maginnis, F. J.: Differential Analyzer Applications, *Gen. Elec. Rev.*, **48**(5):54–59 (May, 1945).
101. Cook, A. C., and F. J. Maginnis: More Differential Analyzer Applications, *Gen. Elec. Rev.*, **52**(8):14–20 (August, 1949).
102. Meriam, J. L.: Differential Analyzer Solution for the Stresses in a Rotating Bell-shaped Shell, *J. Franklin Inst.*, **250**:115–133 (1950).
103. Hartree, D. R., and J. R. Womersley: A Method for the Numerical or Mechanical Solution of Certain Types of Partial Differential Equations, *Proc. Roy. Soc.* (*London*), A, **161**:353–366 (1937).

PART 2: A-C/ELECTROMECHANICAL ANALOG COMPUTERS

By GRANINO A. KORN

SURVEY

8.2.1. A-C/Electromechanical Analog Computers. Representation of Variables. A-C/electromechanical analog computers are used as special-purpose computers and as integral parts of special control systems (navigation and fire-control computers, some flight simulators). A-C/electromechanical computers, in general, are not used to obtain numerical answers or design data in quantitative studies like the computing machines forming the main topic of this handbook. Only a brief review will be given here.

A-C analog computing elements represent variables by amplitudes of fixed-phase sinusoidal or square-wave a-c carriers (most frequently 60, 400, or 1,000 cps); complete (180-deg) phase reversal corresponds to a sign change. Electromechanical computing elements (a-c servomechanisms) have a-c inputs and a-c and shaft-position outputs.

Some special-purpose computers represent magnitudes and azimuth angles of two-dimensional *vectors* by carrier amplitudes and phases.

8.2.2. Computing Elements. A-C/electromechanical analog computers implement

Addition and subtraction by means of (a-c) summing amplifiers (Sec. 2.2), by means of series-connected transformer windings, or with mechanical differentials (Sec. 8.1)

Multiplication by constant coefficients by means of potentiometers, resistive networks, transformers, autotransformers, and/or a-c amplifiers

Multiplication and division of variables by means of a-c computer servomechanisms (a synchro transformer can replace the follow-up potentiometer)

Multiplication by trigonometric functions by means of induction resolvers (Fig. 8.2.1)

Generation of functions by servo-driven function potentiometers, switches, or autotransformers

Integration with respect to time by means of integrating servos (rate-feedback servomechanisms, Fig. 8.2.2), by means of integrating gyros, or by demodulation and d-c integration (Sec. 2.2)

Differentiation with respect to time by means of servo-driven induction generators or rate gyros; or by demodulation and d-c differentiation (Sec. 2.2)

For all these operations, a-c/electromechanical computing elements permit component accuracies between 0.1 and 1 per cent of full scale for suitable applications and signal frequency ranges. In addition, a-c/electromechanical computers can comprise mechanical computing elements, such as gears, differentials, cams, linkages, mechanical integrators, etc. (Sec. 8.1), and heat-transfer computing elements (Sec. 8.3).

8.2.3. Design of A-C/Electromechanical Computing Elements: References. The design of accurate a-c/electromechanical computing elements involves the entire field of a-c servomechanisms, amplifiers, and control systems, specifically

Design of gears, gear trains, and bearings
Clutches and brakes
Synchros, induction resolvers, and a-c amplifiers
Fractional-horsepower a-c motors and complete a-c servomechanisms
Tachometers and induction generators
Accelerometers and gyros

A comprehensive and practical handbook-type review of these topics forms fully one-third of Truxal's *Control Engineers' Handbook* (Ref. 1, Secs. 6 through 14 and 17). Additional references are listed at the end of this section.

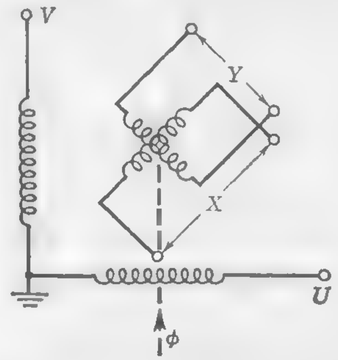


FIG. 8.2.1. Use of an induction resolver to compute $X = U\cos\phi + V\sin\phi$, $Y = -U\sin\phi + V\cos\phi$. A transistorized resolver-driving a-c amplifier can be mounted on the resolver case itself.

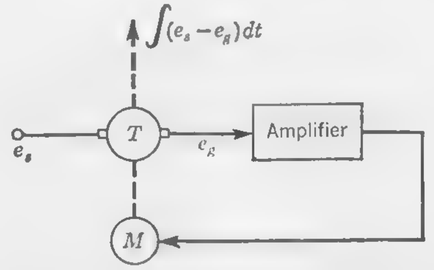


FIG. 8.2.2. A-C integrating servomechanism. The a-c servomotor M and the a-c excited induction generator T may be assembled as a single unit. Output speed is proportional to input voltage; the unit may be calibrated with the aid of a stroboscopic disk on the motor axis.

A-C VS. D-C ANALOG COMPUTERS

8.2.4. Advantages of A-C Analog Computation. The essential advantage of a-c analog computation is *the absence of d-c amplifiers with their attendant drift problems and multiple power supplies.* Again, induction resolvers driven by a-c integrating servos can rotate indefinitely without scale-factor limitations. Although induction resolvers must incorporate slip rings, their life expectancy exceeds that of sine-cosine potentiometers. The possibility of transformer-coupled interconnections adds to circuit flexibility. Above all, a single servomechanism driving multiple potentiometers and/or resolvers can produce accurate products of 5 to 20 input functions by either the servo input or its time integral. A-C computer servomechanisms do not require choppers or motor brushes, use simple circuits and power supplies, and are easily transistorized.

8.2.5. Disadvantages. Unlike d-c analog computers, *a-c analog computers do not permit straightforward all-electrical integration and differentiation.* Although special filter networks permit derivative and/or integral control in a-c servos,[1] this technique is not suitable for accurate computation, and demodulation is required for all-electrical integration or differentiation. Integrating servomechanisms tend to be less accurate than d-c integrators, and servo frequency response is limited.

Accurate computation with a-c carrier amplitudes requires *precise control of carrier phase* in circuits which may involve wiring capacitances and inductive transformers, synchros, and resolvers. As a result of this, an entire technology of impedance-matching amplifiers, phase-correction networks, and quadrature-elimination circuits has evolved; these circuits must be accurately adjusted and temperature-compensated to prevent phase-shift errors.

8.2.6. Conclusions. The considerations of Arts. 8.2.4 and 8.2.5 indicate that a-c analog-computing elements are not suitable for multipurpose differential analyzers but can be eminently useful in *special-purpose computers* requiring continued drift-free

trigonometrical computations, especially in quantity-produced air-borne navigation and fire-control computers. The frequency response of 400-cps servos, if not of 60-cps servos, is adequate for many applications. In spite of the maintenance problems possibly associated with moving parts, maintenance as well as initial costs of a-c servos driving multiple-function generators (Art. 8.2.4) are very hard to beat with all-electronic components of comparable accuracy.[5] Nevertheless, advent of reliable inexpensive chopper-stabilized d-c amplifiers (Sec. 2.3), and particularly of all-solid-state devices of this type (Secs. 7.1 and 7.2), has cut into the market for a-c analog computers, e.g., in the field of large flight simulators (Sec. 5.6). Finally, compact and flexible airborne digital computers increasingly replace separate electromechanical navigation and fire-control computers.

References

1. Truxal, John G.: *Control Engineers' Handbook*, McGraw-Hill Book Company, Inc., New York, 1958 (note bibliographies).
2. Greenwood, I. A., Jr., J. V. Holdam, Jr., and D. Macrae, Jr. (eds.): *Electronic Instruments*, McGraw-Hill Book Company, Inc., New York, 1948.
3. Ledgerwood, B. K.: *Control Engineering Manual*, McGraw-Hill Book Company, Inc., New York, 1957.
4. Gibson, J. E., and F. B. Tuteur: *Control System Components*, McGraw-Hill Book Company, Inc., New York, 1958.
5. Connelly, M. E.: Computers for Aircraft Simulation, *Report 7591-R-2*, Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., 1959.

PART 3: HEAT-TRANSFER COMPUTING ELEMENTS

By PAUL SAVET

INTRODUCTION

8.3.1. Heat-transfer Computing Elements. Heat-transfer computing elements perform basic operations of algebra and calculus on amplitude-modulated a-c voltages as well as on d-c voltages. No moving parts are involved; accuracies between 0.4 and 1 per cent may be obtained over a dynamic range of 500 to 1 for signal frequencies (modulation frequencies) up to at least 10 cps. While input and output variables are electrical signals, the intermediate variable physically employed is the *rate of heat transfer* between wire-pair elements. Hence computer operation does not depend on the carrier frequency (60 to 1,000 cps) of amplitude-modulated signals; in particular, differentiation and integration with respect to time do not require d-c amplifiers with their attendant drift and power-supply problems.

Heat-transfer computing elements are small and relatively inexpensive. They are insensitive to ambient-temperature variations as well as to shock and vibration. Disadvantages arise from difficulties in producing matched heat-transfer elements, and from inherent time delays. The essential features of the heat-transfer technique were developed at the Arma Division of the American Bosch Arma Corporation by J. Statsinger and, more recently, by G. M. Davidson and others.

OPERATING PRINCIPLES

8.3.2. Heat-transfer Transducers. In the basic heat-transfer transducer (Fig. 8.3.1*a*), a bifilar *heating element* (No. 40 Teflon-insulated nichrome wire) and a *sensing element* or sensing elements (No. 40 Teflon-insulated nickel wire, large temperature coefficient of resistance) are wound together on a pyrex rod. Two or more such units may be mounted in a miniature-tube envelope; no vacuum is required. Referring to Fig. 8.3.1*a*, a gain-changing voltage v is added to the heater voltage e; the resulting temperature change varies the resistance of the sensing element. In the circuit of Fig. 8.3.1*b*, the output voltage x_0 is determined by differential resistance changes of the

upper and lower sensing elements. Referring to Fig. 8.3.1*b*, the difference in the heat-transfer rates is proportional to the time average of $(e + v)^2 - (e - v)^2$, and hence to $EV \cos \vartheta$; in many applications, $\vartheta = 0$. If the signal y applied to the sensing elements is sufficiently small to prevent "self-heating," the rms output voltage X_0 is roughly given by the linear approximation

$$X_0 = \frac{k}{Tp + 1} V \qquad \left(p \equiv \frac{d}{dt}\right) \tag{8.3.1}$$

if V and E are sufficiently small (Fig. 8.3.1*c*). k is the transducer attenuation constant, and T is the transducer time constant determined by the heat-transfer lag. In

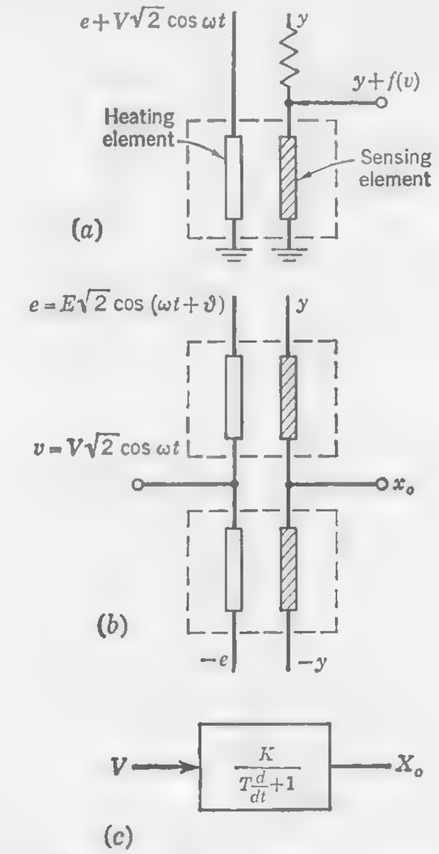


FIG. 8.3.1. Principle of the heat-transfer transducer. The approximate transfer function of Fig. 9.3.1*c* relates input and output rms values.

practice, k is between 0.1 and 0.2, and T is at least 0.2 sec. Transducer operating temperatures are of the order of 100°C.

8.3.3. The Heat-transfer Multiplier-divider. In Fig. 8.3.2, heat-transfer transducers with two pairs of sensing elements are used as voltage-sensitive variable-gain transducers in a feedback-multiplier circuit (see also Sec. 3.2). The output voltage is given by

$$x_0 = \frac{X_1 X_2}{Y} \frac{Ak}{1 + Ak} \frac{1}{(Tp/1 + Ak) + 1} \qquad \left(p \equiv \frac{d}{dt}\right) \tag{8.3.2}$$

The feedback linearizes the gain-setting operation and reduces the effective time constant; in practice, the attainable loop gain is limited by considerations of stability and ampli-

fier saturation. The feedback amplifier, which may be transistorized, must supply about 1 watt of power.

8.3.4. Approximate Differentiation and Integration. Since the performance equation (1) applies to rms rather than instantaneous voltages, the heat-transfer technique

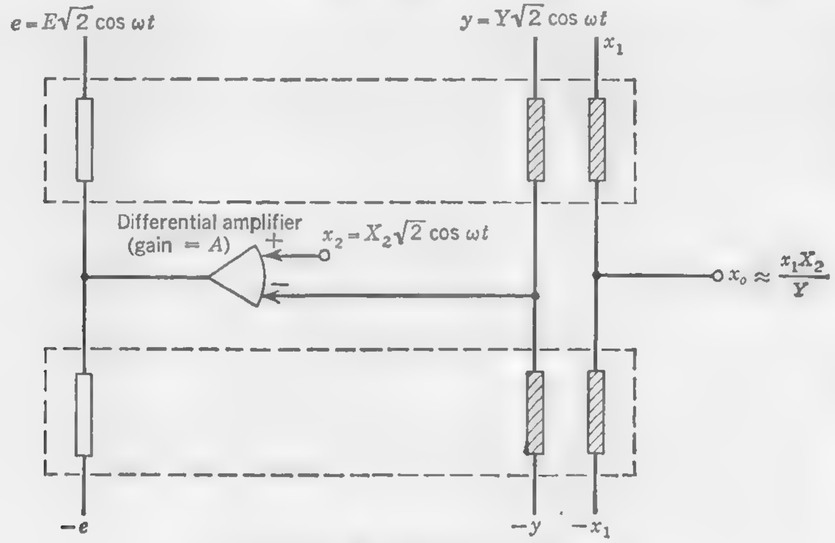


FIG. 8.3.2. Heat-transfer multiplier.

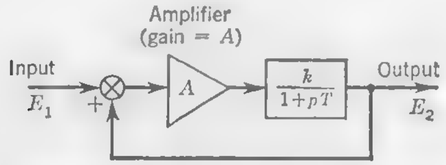


FIG. 8.3.3. Heat-transfer integrator. Feedback yields the transfer function

$$\frac{E_2}{E_1} = \frac{Ak}{1 - Ak} \frac{1}{pT/(1 - Ak) + 1}$$

which corresponds to approximate integration. Effective integrator time constants $T/(1 - Ak)$ up to 30 min have been obtained.

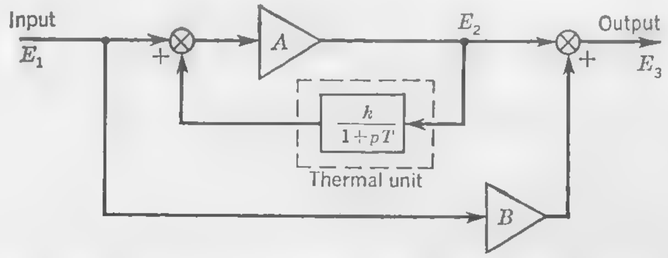


FIG. 8.3.4. Feedback circuit for approximate differentiation.

permits differentiation and integration of amplitude-modulated a-c signals in the manner of Figs. 8.3.3 and 8.3.4. Figures 8.3.5 to 8.3.8 illustrate a number of other applications.

8.3.5. Self-heating Transducers. In the self-heating transducer of Fig. 8.3.9, the heating elements L_1, L_2 are identical with the sensing elements; physically, L_1 and L_2 are special tungsten-filament lamps requiring about 1 watt driving power.

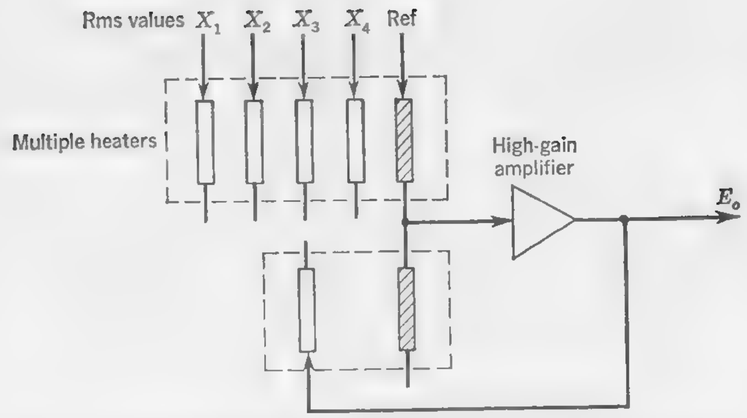


FIG. 8.3.5. Root-mean-square adder circuit with multiple heating elements.

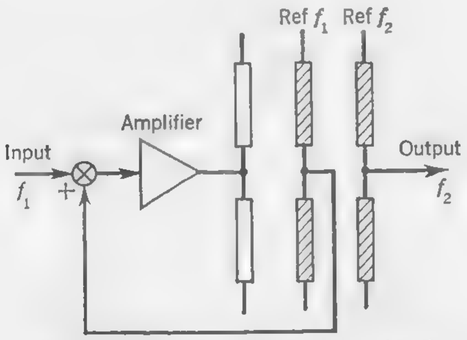


FIG. 8.3.6. Frequency-converter circuit.

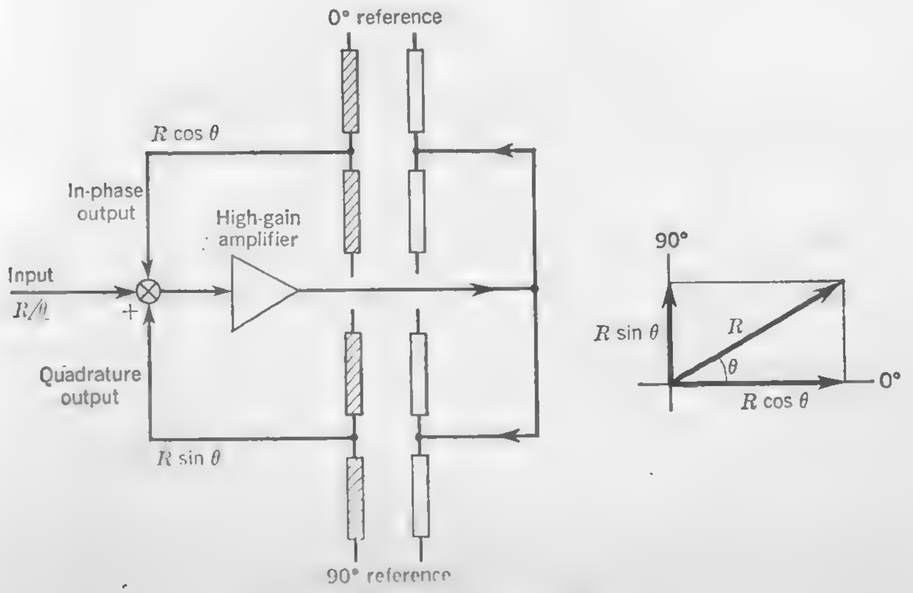


FIG. 8.3.7. An a-c resolver circuit.

The reference signals shown are 180 deg out of phase; thus, while L_1 is heated, L_2 is sampled by the differential-transducer circuit, and vice versa. The high filament temperatures used reduce the open-loop time constant T to a few milliseconds; the

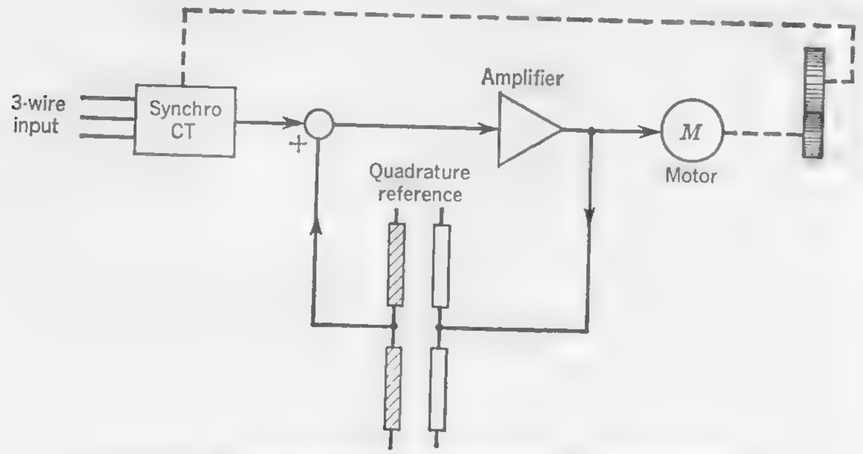


FIG. 8.3.8. Quadrature-rejection circuit for a servomechanism.

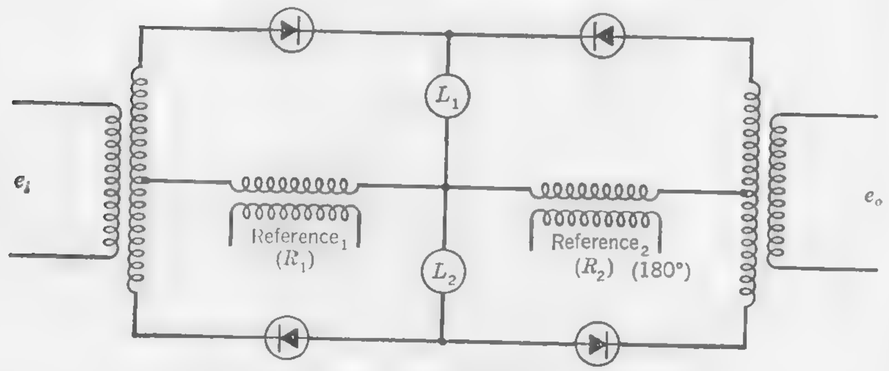


FIG. 8.3.9. A self-heating thermal transducer.

attenuation constant corresponds to about 6 db. The output waveform of the transducer shown in Fig. 8.3.9 must be added to a second full-wave rectified signal of the same frequency and phase to produce a sinusoidal output voltage.

References

1. Davidson, G. M.: Thermal Elements Simplify Computing Circuits, *Elec. Mfg.*, June, 1954.
2. Davidson, G. M., W. Djinis, and P. H. Savet: Subminiature Thermal Computing Elements, *Electronic Equipment*, June, 1955.
3. Gedney, G.: Thermal Computers, *Arma Eng.*, **1**(4), (January, 1958); **1**(5), (March, 1958).
4. Davidson, G. M., and G. Gedney: Thermal Computers, *Arma Eng.*, **1**(6), (June, 1958); **2**(1), (August, 1958).

Section 9

NETWORK-TYPE DIRECT-ANALOGY COMPUTERS AND FIELD-PROBLEM ANALOGIES

By

DONALD T. GREENWOOD, *Associate Professor of Instrumentation Engineering, University of Michigan, Ann Arbor, Mich.*

WILLIAM J. DIXON, *President, Computer Engineering Associates, Inc., Pasadena, Calif.*

R. P. WASHBURN, *Analyst, Marketing Research, General Electric Company, Computer Department, Phoenix, Ariz.*

WALTER J. KARPLUS, *Associate Professor of Engineering, University of California, Los Angeles, Calif.*

WALTER W. SOROKA, *Professor of Mechanical Engineering, University of California, Berkeley, Calif.*

CONTENTS

PART 1: NETWORK ANALOGIES OF DYNAMICAL SYSTEMS

Donald T. Greenwood

Introduction

9.1.1. The Direct-analogy Computer 9–3

Representation of Basic Mechanical Systems

9.1.2. Loop and Nodal Analogies of Mechanical Elements..... 9–3
9.1.3. Loop and Nodal Analogies for a Mass-spring-damper System 9–3
9.1.4. The Scaling Equations..... 9–5
9.1.5. Realizability of Passive-network Analogies..... 9–7
9.1.6. Coordinate Transformations.. 9–8

Dynamical Structural Analogies

9.1.7. The Torsion Rod..... 9–12
9.1.8. Beams in Bending..... 9–12
9.1.9. Beam Boundary Conditions.. 9–15
9.1.10. Beams with Rotary Inertia.. 9–15
9.1.11. The Russell Beam Analogy.. 9–15
9.1.12. Coupled Bending and Torsion of a Beam..... 9–17
9.1.13. The Three-beam Circuit..... 9–18

Analogies for Static Structures

9.1.14. A Comparison of Static and Dynamic Circuits..... 9–19
9.1.15. The Plane-stress Circuit..... 9–20
9.1.16. The Plate Analogy..... 9–23
9.1.17. The Ring Analogy..... 9–26
9.1.18. The Analogy for a Stiffened Cylindrical Shell..... 9–29

Errors

9.1.19. Types of Errors..... 9–30
9.1.20. Finite-difference Approximations..... 9–30
9.1.21. Parasitic Impedances..... 9–34

PART 2: DIRECT-ANALOGY COMPUTERS: COMPONENTS AND SYSTEMS

William J. Dixon

Introduction

9.2.1. General Requirements of a Direct-analogy Computer.... 9–36

Passive Elements

9.2.2. Selection of Frequency Range, Impedance Level, and Power Level..... 9–37
9.2.3. Transformers and Inductors.. 9–41
9.2.4. Resistors..... 9–44
9.2.5. Capacitors..... 9–44

Active Elements and Nonlinear Elements

9.2.6. Active Elements in General.. 9–45
9.2.7. Amplifiers..... 9–45
9.2.8. Multipliers and Function Generators..... 9–49
9.2.9. Special Nonlinear Elements.. 9–49

Peripheral Equipment

9.2.10. Excitation..... 9–51
9.2.11. Metering..... 9–54
9.2.12. Control Console..... 9–58

Operation

9.2.13. Operating Features..... 9–59
9.2.14. Checking..... 9–59
9.2.15. Accuracy..... 9–59
9.2.16. Typical Installations..... 9–60

PART 3: POWER-SYSTEM NETWORK ANALYZERS

R. P. Washburn

Introduction

9.3.1. Power-system Network Analyzers..... 9–61
9.3.2. D-C and Direct-analog A-C Network Analyzers..... 9–61
9.3.3. Modified-analog Network Analyzers..... 9–61

Design of Network Analyzers

9.3.4. Network-analyzer Components..... 9–62
9.3.5. Base Quantities and Scaling. 9–63
9.3.6. Design Details..... 9–63
9.3.7. Semiautomatic Programming and Read-out..... 9–67

Solution of Network Problems

9.3.8. Required Data..... 9–67
9.3.9. Types of Problems Solved... 9–68
9.3.10. Fault Studies..... 9–68

PART 4: SPECIAL METHODS FOR THE SOLUTION OF FIELD PROBLEMS

Walter J. Karplus

Resistance Networks

9.4.1. The Resistance-network Analogy..... 9–69
9.4.2. Laplace's Equation in Rectangular Cartesian Coordinates.. 9–69
9.4.3. Laplace's Equation in Curvilinear Coordinates..... 9–72
9.4.4. Asymmetrical Networks..... 9–76
9.4.5. Analog for Equations of the Poisson Type..... 9–76
9.4.6. Analog for the Diffusion Equation..... 9–77
9.4.7. Analog for the Wave Equation..... 9–78
9.4.8. Analog for the Biharmonic Equation..... 9–79

Conductive-solid Analogies

9.4.9. Principle of the Analogy..... 9–80
9.4.10. Resistance Paper–Teledeltos. 9–80
9.4.11. Conductive Rubbers..... 9–81
9.4.12. Miscellaneous Conductive Models..... 9–81
9.4.13. Analog for Poisson's Equation..... 9–81

Conductive Liquids—The Electrolytic Tank

9.4.14. Principle of Operation..... 9–82
9.4.15. Design of Electrolytic Tanks. 9–82

Stretched-membrane Analogies

9.4.16. Membrane Analogies..... 9–84

PART 5: HYDRODYNAMIC ANALOGIES

Walter W. Soroka

Analogies for Two-dimensional Poisson and Laplace Fields

9.5.1. Introduction..... 9–85
9.5.2. Hydrodynamic Analogies for Prismatic Bars in Torsion.... 9–85
9.5.3. Sandbed Mappers..... 9–85
9.5.4. Hydrodynamic Analogy for Shear Stress in Bending..... 9–86
9.5.5. Hydrodynamic Analogies for Other Field Problems..... 9–86

Hydrodynamic Representation of Creep and Relaxation in Complex Materials

9.5.6. Representation of Creep and Relaxation..... 9–86

A Hydraulic Analogy for Transonic Aerodynamics

9.5.7. A Hydraulic Analogy for Transonic Flow..... 9–87

PART 1: NETWORK ANALOGIES OF DYNAMICAL SYSTEMS

By Donald T. Greenwood

INTRODUCTION

9.1.1. The Direct-analogy Computer. The direct-analogy computer is distinguished by its use of passive circuit elements (capacitors, resistors, inductors, and transformers) as fundamental elements in the computing process. A network composed of these elements can be described in terms of a system of ordinary linear differential equations with constant coefficients. It is known that many systems involving other physical processes (mechanical vibrations or heat flow, for example) are also described or closely approximated by equations of the same form. Often it is possible to represent a system of this sort by an "analogous" passive electrical network. The utility of an electrical analogy or model lies in the ease with which system parameters may be changed and the convenience in making measurements. Also, it turns out that the computer representation is relatively easy to calculate and to set up for many physical systems of interest.

Although the direct-analogy computer is often used to analyze linear passive systems, it is not restricted to such use. Active elements such as high-gain d-c amplifiers, multipliers, and function generators are often used with the computer. Such devices increase its flexibility and enable one to analyze, in addition, many systems which are unstable or nonlinear (see also Sec. 9.2).[8]

REPRESENTATION OF BASIC MECHANICAL SYSTEMS

9.1.2. Loop and Nodal Analogies of Mechanical Elements. There are two general types of electrical-network analogies for mechanical elements; namely, (1) the *loop* or *force-voltage* analogy, and (2) the *nodal* or *force-current* analogy. The basic correspondences involved in the loop and nodal analogies are shown in Table 9.1.1. By comparing the equations describing the various electrical and mechanical elements, the correspondences shown at the bottom of the table can be deduced.

All the passive-network representations for mechanical systems that are shown on following pages can be considered as exact representations (within the limitations of component accuracy) for lumped mechanical systems. These lumped mechanical systems comprise masses, springs, dampers, and levers, plus their rotational counterparts. Additional transformers are used for coordinate transformations.

9.1.3. Loop and Nodal Analogies for a Mass-spring-damper System. The differential equation for the mechanical system of Fig. 9.1.1*a* is

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = f(t) \tag{9.1.1}$$

where the displacement x is measured from the equilibrium position.

The nodal analogy for this system is shown in Fig. 9.1.1*b* and is described by the equation

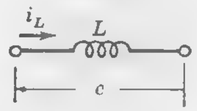
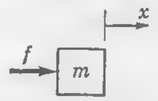
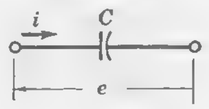
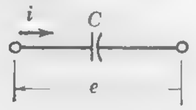
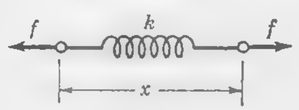
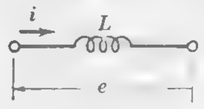
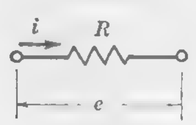
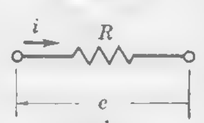
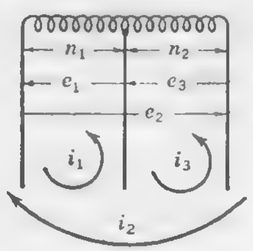
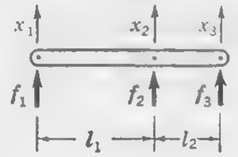
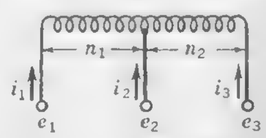
$$C\frac{de}{dt} + \frac{e}{R} + \frac{1}{L}\int_{t_0}^{t} e\,dt = i(t) \tag{9.1.2}$$

where the inductor current is assumed to be zero at time $t = t_0$. Noting again that

$$\frac{dx}{dt} = v$$

we see that the two systems are indeed analogous, with the correspondences given in Table 9.1.1.

Table 9.1.1. Loop and Nodal Analogies for Basic Mechanical Elements

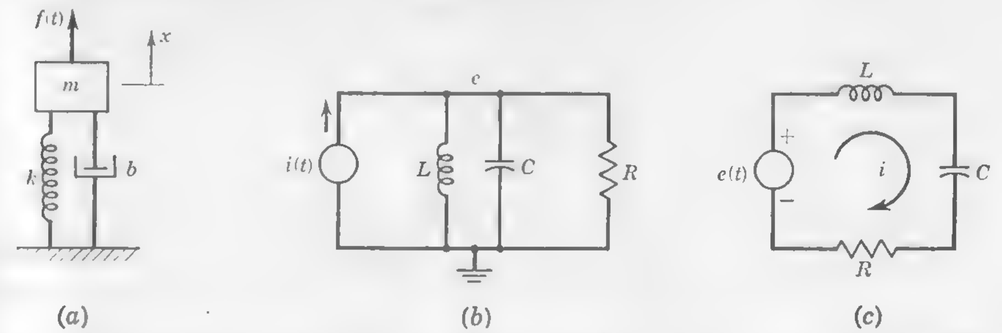
Loop analogy	*Mechanical element*	*Nodal analogy*
Inductor  $e = L\frac{di}{dt}$	Mass  $v = \frac{dx}{dt}$ $f = m\frac{d^2x}{dt^2} = m\frac{dv}{dt}$	Capacitor  $i = C\frac{de}{dt}$
Capacitor  $e = \frac{1}{C}\int_{t_0}^{t} i\,dt + e_0$ where $e(t_0) = e_0$	Spring  $f = k\int_{t_0}^{t} v\,dt + f_0$ where $f(t_0) = f_0$	Inductor  $i = \frac{1}{L}\int_{t_0}^{t} e\,dt + i_0$ where $i(t_0) = i_0$
Resistor  $e = Ri$	Damper $f = bv$	Resistor  $i = \frac{1}{R}e$
Autotransformer  $e_1 + e_2 + e_3 = 0$ $\frac{e_1}{n_1} = \frac{e_3}{n_2}$ $n_1(i_2 - i_1) = n_2(i_3 - i_2)$	Massless lever  $f_1 + f_2 + f_3 = 0$ $l_1 f_1 = l_2 f_3$ $\frac{v_2 - v_1}{l_1} = \frac{v_3 - v_2}{l_2}$	Autotransformer  e_1, e_2, e_3 are voltages relative to ground. n_1, n_2 are the number of turns in portions of the winding. $i_1 + i_2 + i_3 = 0$ $n_1 i_1 = n_2 i_3$ $\frac{e_2 - e_1}{n_1} = \frac{e_3 - e_2}{n_2}$

Analogous Quantities

Loop analogy	*Mechanical system*	*Nodal analogy*
Voltage e	Force f	Current i
Current i	Velocity v	Voltage e
Inductance L	Mass m	Capacitance C
Elastance $\frac{1}{C}$	Spring stiffness k	Inverse inductance $\frac{1}{L}$
Resistance R	Damper constant b	Conductance $\frac{1}{R}$
Inverse turns $\frac{1}{n}$	Lever length l	Turns n

Equation (1) states that the sum of the forces (including inertial forces) acting in a given direction at a point is equal to zero. This is essentially a statement of Newton's second law. Similarly, Eq. (2) states that the total current entering a node is equal to zero. This is Kirchhoff's current law.

It is important to note the *topological similarity* between the mechanical system and its nodal analogy. This similarity in form may not be apparent at first glance. One must observe that each mechanical element is in reality a two-terminal element. The mass is "connected" between the node x and ground since its acceleration is referred to an inertial system, which is the earth in this case. Also, the force $f(t)$ is the output from a force generator which is assumed to be pushing against the earth or an object fixed with respect to it. Thus each system consists of a parallel connection of four two-terminal elements between a single node and ground.



(*a*) (*b*) (*c*)

FIG. 9.1.1. Passive electrical analogies for a mass-spring-damper system. (*a*) The mechanical system. (*b*) The nodal analogy. (*c*) The loop analogy.

Consider now the loop analogy (Fig. 9.1.1*c*) for the mass-spring-damper system. The differential equation describing this circuit is

$$L\frac{di}{dt} + Ri + \frac{1}{C}\int_{t_0}^{t} i\,dt = e(t) \tag{9.1.3}$$

where the charge on the capacitor is assumed to be zero at $t = t_0$. Again, by comparing Eqs. (1) and (3), we note that the correspondences apply as given by Table 9.1.1 for the loop analogy.

The loop analogy is not topologically similar to the mechanical system. This is an important disadvantage of the loop analogy compared with the nodal analogy and is perhaps the principal reason for its infrequent use.

In general, the loop and nodal representations of a given mechanical system will be electrical *duals*. This means, among other things, that the roles of loops and nodes are interchanged. Thus a given nodal velocity of the mechanical system will correspond to a node voltage in the nodal analogy and to a loop current in the loop analogy.

In the material which follows, the *nodal analogy will be used exclusively.*

9.1.4. The Scaling Equations. Thus far we have been concerned primarily with the form of the electric circuit and little attention has been paid to numerical values of the elements. Now we shall obtain scaling equations whereby numerical values may be obtained.

The differential equations of motion for a lumped mechanical system with n degrees of freedom can be written in the form

$$\sum_{j=1}^{n} (m_{ij}p^2 + b_{ij}p + k_{ij})x_j = f_i(t) \qquad (i = 1, 2, \ldots, n) \tag{9.1.4}$$

where the differential operator is denoted by p and $f_i(t)$ is the generalized force corresponding to the ith coordinate.

Similarly, the nodal electric-circuit equations can be written

$$\sum_{j=1}^{n} \left(C_{ij}\bar{p}^2 + \frac{1}{R_{ij}}\bar{p} + \frac{1}{L_{ij}} \right) \frac{e_j}{\bar{p}} = i_i(\tau) \qquad (i = 1, 2, \ldots, n) \qquad (9.1.5)$$

where τ is the electric circuit or computer time and $\bar{p}$ is the corresponding differential operator.

The time-scale factor N relates the actual physical time t to the computer time τ.

$$t = N\tau \qquad (9.1.6)$$

For the usual mechanical system, N is a number larger than 1. In other words, the observed computer frequencies are greater than the corresponding frequencies of the physical system. A typical problem might show the lowest significant frequencies in the computer solution to be the order of 100 cps whereas the corresponding mechanical frequencies might be ten to a hundred times smaller.

From Eq. (6) we obtain the relationship

$$\bar{p} = Np \qquad (9.1.7)$$

The mechanical velocities are related to the corresponding voltages by the equation

$$px_j = \frac{K}{N} a_j e_j \qquad (j = 1, 2, \ldots, n) \qquad (9.1.8)$$

where the a_j are the voltage scale factors. It is usually not necessary or advisable to choose a different scale factor at each node point, i.e., for each value of j. Often a single voltage scale factor is sufficient.

The scale factor K relates the energy levels of the two systems. The mechanical-system energy level is K^2 times that in the electrical system. The factor K does not influence the value of any electrical component. Rather, it influences the magnitude of the forcing current (or voltage) and also, of course, the voltage and current levels throughout the circuit. Often, however, relative rather than absolute magnitudes are desired and the scale factor K can be ignored.

The forcing current is given by

$$i_i(\tau) = \frac{a_i f_i(t)}{K} \qquad (i = 1, 2, \ldots, n) \qquad (9.1.9)$$

Substituting Eqs. (7), (8), and (9) into the mechanical-system equation (4), we obtain

$$\sum_{j=1}^{n} a_i a_j \left(m_{ij} \frac{\bar{p}^2}{N^2} + b_{ij} \frac{\bar{p}}{N} + k_{ij} \right) \frac{e_j}{\bar{p}} = i_i(\tau) \qquad (i = 1, 2, \ldots, n)$$

which, when compared with Eq. (5), yields the following:

$$C_{ij} = \frac{1}{N^2} a_i a_j m_{ij} \qquad (9.1.10)$$

$$\frac{1}{R_{ij}} = \frac{1}{N} a_i a_j b_{ij} \qquad (9.1.11)$$

$$\frac{1}{L_{ij}} = a_i a_j k_{ij} \qquad (9.1.12)$$

Equations (7) through (12) are the basic scaling equations for the direct-analogy computer.

9.1.5. Realizability of Passive-network Analogies. Equations (10), (11), and (12) are the principal scaling equations that are used to set up a passive-network analogy for a lumped mechanical system. For a general lumped mechanical system with n coordinates the subscripts i and j will take on all values from 1 to n. The nodal analogy will have n nodes (plus ground) and $n(n + 1)/2$ C's, R's, and L's each, in the general case. It is important to note that a capacitance C_{ii} as calculated by Eq. (10) is *not* the value of the actual capacitor connected from the ith node to ground. Rather, it is the total self-capacitance at the ith node, i.e., the sum of all capacitances connected to the ith node.

The C_{ij} as calculated from Eq. (10) for the case $i \neq j$ must all be negative if sign-changing transformers are to be avoided in the circuit. Each positive C_{ij} will require a sign-changing transformer. These transformers will change the sign of the corresponding mutual capacitances but will not change the sign of the contribution to the self-capacitance. In any event, the size of the capacitor between the ith and jth nodes is $|C_{ij}|$.

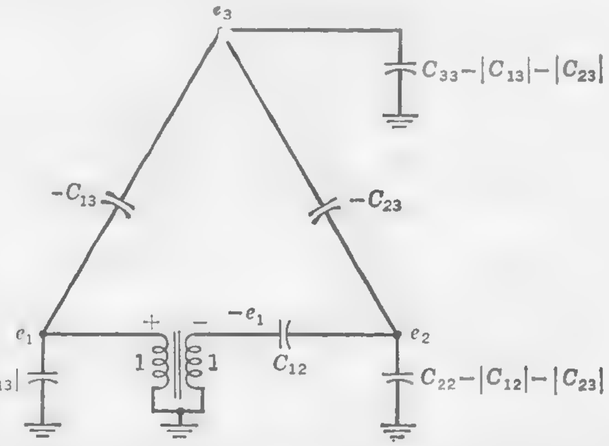


FIG. 9.1.2. An example of inertial representation by a nodal analogy.

To clarify this point, consider the inertial representation shown in Fig. 9.1.2 for the case of three coordinates. The mutual capacitances C_{13} and C_{23} are negative, but C_{12} is positive. Because C_{12} is positive a sign-changing transformer is placed in series with the capacitor.

Note that the condition that the capacitor from the ith node to ground be positive is that

$$C_{ii} - \sum_{j \neq i} |C_{ij}| \geq 0 \qquad (9.1.13)$$

What has been stated with regard to restrictions on the C's for physical realizability is equally true for the L's and R's. If we designate the admittance of any *one* of these element types by Y_{ij}, we can summarize the physical realizability conditions as follows:

1. $Y_{ij} = Y_{ji}$
2. $Y_{ii} \geq \sum_{j \neq i} |Y_{ij}| \qquad (i = 1, 2, \ldots, n)$

It should be emphasized that these conditions need not apply to the m_{ij}, b_{ij}, and k_{ij}, provided that a linear coordinate transformation can be found such that the conditions are met in the new coordinate system. These linear transformations are performed in the circuit by the use of transformers.

It has been stated that one of the principal advantages of the nodal analogy is that a topological similarity exists between the mechanical and electrical systems. However,

we can see from the scaling equations that this is strictly true only for the case of a uniform scale factor throughout the circuit. Even so, for practical cases, there is generally enough uniformity of scale factors that the form of the nodal circuit can be obtained easily.

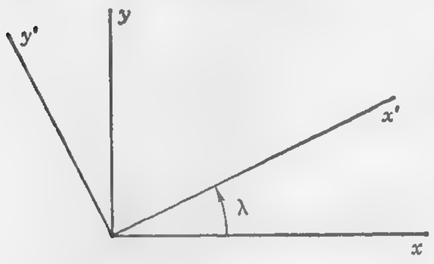


FIG. 9.1.3. A rotation of axes.

9.1.6. Coordinate Transformations. Coordinate transformations are used for several reasons. Among these are (1) to make possible the representation of an otherwise impossible system, (2) to simplify the scaling, (3) to provide convenient coordinates for representing complicated structures, (4) to change impedance levels in order to improve the accuracy of computation, and (5) to impose constraints on the system.[7]

As a first example of a coordinate transformation, consider the rotation of axes shown in Fig. 9.1.3. The transformation equations are

$$x' = x \cos \lambda + y \sin \lambda \tag{9.1.14}$$
$$y' = -x \sin \lambda + y \cos \lambda \tag{9.1.15}$$

This transformation can be accomplished using the transformer circuit of Fig. 9.1.4*a*, where unity scale factors have been assumed. This circuit requires four transformers

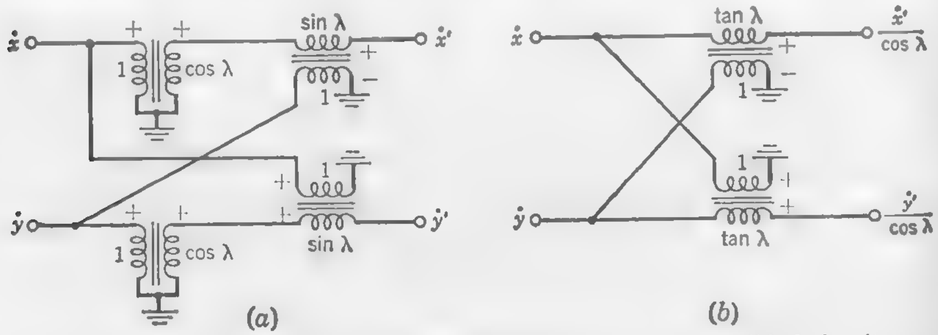


FIG. 9.1.4. (*a*) Circuit for coordinate-axis rotation. (*b*) Simplified circuit.

and would prove costly in terms of transformers if used very many times in a given computer setup. Another representation is suggested by dividing Eqs. (14) and (15) by cos λ, giving

$$\frac{x'}{\cos \lambda} = x + y \tan \lambda \tag{9.1.16}$$
$$\frac{y'}{\cos \lambda} = -x \tan \lambda + y \tag{9.1.17}$$

This circuit is shown in Fig. 9.1.4*b*. Note that the voltage scale factor has now been changed for the primed system. In case λ lies between $\pi/4$ and $\pi/2$ it is preferable to divide equations by sin λ rather than cos λ. This will keep the scale-factor ratio less than $\sqrt{2}:1$.

A rotation-of-axes transformation is particularly useful at junctions or bends in structural members. It enables each section of the over-all structural representation to have its local coordinate system with coordinate transformations used at the junctions.

An important feature of transformers as coordinate transforming devices is that, ideally, the power input and output are at all times equal. In other words, a certain voltage ratio between primary and secondary windings implies the inverse current ratio between these windings, and vice versa. Thus only the displacement (or velocity)

equations need be considered directly in finding a transformation circuit. The implied force-transformation equations will automatically be enforced by the circuit.

A second type of coordinate transformation is accomplished by a two-winding transformer whose turns ratio is different from +1. This type of transformation is used either to obtain a realizable circuit or for reasons of improved scaling and better accuracy.

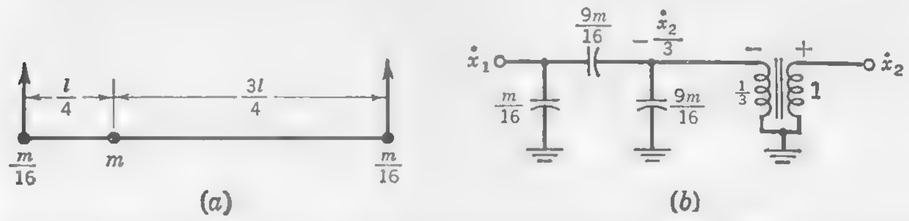


FIG. 9.1.5. A rigid bar with point masses, and the use of a coordinate transformation in its representation.

As an example of this sort of transformation consider the rigid bar with point masses shown in Fig. 9.1.5*a*. In terms of the given coordinates, the inertial coefficients are

$$m_{11} = \frac{10}{16} m \qquad m_{12} = \frac{3}{16} m$$
$$m_{21} = \frac{3}{16} m \qquad m_{22} = \frac{2}{16} m \tag{9.1.18}$$

This inertial description is not directly realizable without using transformers because the mutual mass m_{12} is positive and is larger than the self-mass m_{22}.

If we make the transformation

$$x_1 = x_1'$$
$$x_2 = -3x_2' \tag{9.1.19}$$

we find that the inertial coefficients in the primed coordinate system are

$$m_{11}' = \frac{10}{16} m \qquad m_{12}' = \frac{-9}{16} m$$
$$m_{21}' = \frac{-9}{16} m \qquad m_{22}' = \frac{18}{16} m \tag{9.1.20}$$

Now we can represent the system as shown in Fig. 9.1.5*b*. Unit scale factors have been assumed. Note that the capacitor sizes are the same as though the voltage scale factor were changed by a factor −3. This has the effect of multiplying the mutual masses by −3 and the self-mass m_{22} by $(-3)^2$ as can be seen by comparing Eqs. (18) and (20).

It should be pointed out that the system also could have been represented directly by using an autotransformer with capacitors proportional to the three masses connected to ground from its three taps.

Another type of coordinate transformation that can be used to obtain physical realizability is one in which the unprimed and primed coordinates are equal except for one coordinate. This unprimed coordinate may be expressed as the linear combination of two primed coordinates.

As an example of this transformation, consider a system with inertial coefficients

$$m_{11} = 14 \qquad m_{12} = m_{21} = 9$$
$$m_{22} = 14 \qquad m_{13} = m_{31} = -5 \tag{9.1.21}$$
$$m_{33} = 9 \qquad m_{23} = m_{32} = -8$$

with respect to the unprimed coordinates. Consider now the transformation

$$\begin{aligned} x_1 &= x_1' - x_2' \\ x_2 &= x_2' \\ x_3 &= x_3' \end{aligned} \tag{9.1.22}$$

In order to find the inertia coefficients with respect to the primed system, write the general linear transformation

$$x_i = \sum_{j=1}^{n} \alpha_{ij} x_j' \qquad (i = 1, 2, \ldots, n) \tag{9.1.23}$$

The transformed inertia coefficients are given by

$$m_{ij}' = \sum_{k=1}^{n} \sum_{l=1}^{n} \alpha_{ik}\alpha_{lj} m_{kl} \qquad (i, j = 1, 2, \ldots, n) \tag{9.1.24}$$

This result is more easily expressed using matrix notation. For a coordinate transformation

$$\{x\} = [\alpha]\{x'\} \tag{9.1.25}$$

The transformed inertia matrix is

$$[m'] = [\alpha]'[m][\alpha] \tag{9.1.26}$$

where the prime outside the brackets indicates the transposed matrix

The resulting inertia coefficients are

$$\begin{aligned} m_{11}' &= 14 \qquad m_{12}' = m_{21}' = m_{13}' = m_{31}' = -5 \\ m_{22}' &= 10 \qquad m_{23}' = m_{32}' = -3 \\ m_{33}' &= 9 \end{aligned} \tag{9.1.27}$$

The final circuit is shown in Fig. 9.1.6.

One further coordinate transformation must be mentioned. This is the transformation between ordinary coordinates and principal coordinates.[5,9] This transformation is most useful when one desires to represent a portion of a system in terms of its first few principal (or normal) modes rather than by a detailed representation. The remainder of the system might be assumed to be represented in greater detail in terms of ordinary coordinates, thereby necessitating a coordinate transformation at the junction.

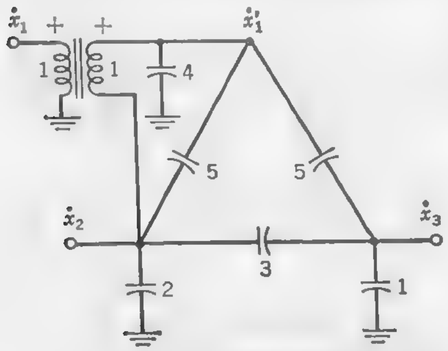


FIG. 9.1.6. A circuit representing the inertial properties given in Eq. (21).

For example, in a vibration study of an airplane wing it might be convenient to represent the fuselage in terms of its principal modes. The wing would be represented in detail using ordinary coordinates in order to permit easy variation of structural parameters and also to obtain the required accuracy. Therefore, a coordinate transformation is necessary at the junction of the wing and fuselage.

Let us consider now an example of a coordinate transformation involving three principal coordinates and two ordinary coordinates. The general transformation

equations are of the form

$$x_j = \sum_{k=1}^{n} A_{jk} U_k \tag{9.1.28}$$

where U_k is the principal coordinate corresponding to the kth mode. The x_j are ordinary coordinates. There are n modes.

An important feature of the description of a system in terms of principal coordinates is that the modes are *orthogonal* or uncoupled. Assuming no damping, each mode can be represented by a parallel inductor and capacitor. If we assume unity scale factors the capacitor and inductor values for representing the kth mode are

$$C_{kk} = \sum_i \sum_j m_{ij} A_{ik} A_{jk} \tag{9.1.29}$$

$$\frac{1}{L_{kk}} = \sum_i \sum_j k_{ij} A_{ik} A_{jk} \tag{9.1.30}$$

The circuit representing Eqs. (28), (29), and (30) is shown in Fig. 9.1.7*a*. This circuit requires six transformers to perform the transformation to ordinary coordinates.

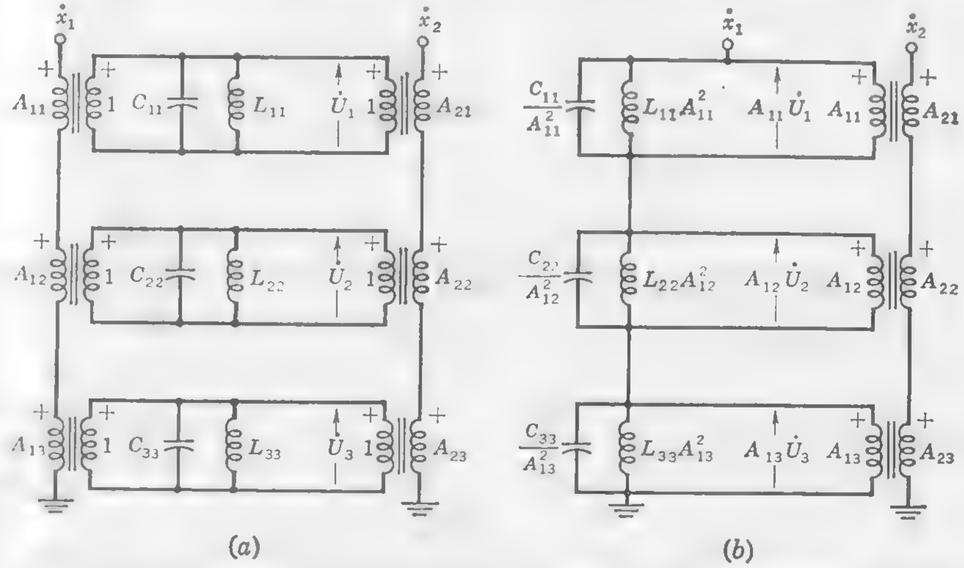


FIG. 9.1.7. Representation of a vibrating system by means of principal modes and employing a coordinate transformation from principal to ordinary coordinates. (*a*) Standard circuit. (*b*) Circuit requiring fewer transformers.

A circuit requiring fewer transformers is shown in Fig. 9.1.7*b*. This circuit was obtained by multiplying the voltage corresponding to $\dot{U}_k$ by the factor A_{1k}. This change of voltage further implies a change in component values. It may be seen that the kth capacitor must be divided by A_{1k}^2 whereas the kth inductor is multiplied by the same factor in order to change the voltage but not the power level. By this means all the left-hand transformers are changed to a one-to-one turns ratio and may be eliminated provided that the individual mode circuits are connected in series. The right-hand transformers now have turns ratios of A_{1k} to A_{2k}.

Another simplification can be made in the circuit representation in terms of principal modes when some natural mode frequencies are much higher or lower than the expected solution frequencies. In this case, the modes of much higher frequency can be approximated in terms of their stiffness (inductance) only, while the modes of much lower frequency are approximated in terms of their mass (capacitance) only.

DYNAMIC STRUCTURAL ANALOGIES

9.1.7. The Torsion Rod. The torsional motion of a straight uniform rod can be obtained by solving the partial differential equation

$$JG\frac{\partial^2\phi}{\partial x^2} = I\frac{\partial^2\phi}{\partial t^2} \tag{9.1.31}$$

which is recognized as the wave equation. The torsional compliance (inverse stiffness) per unit length is $(JG)^{-1}$, ϕ is the torsional displacement, x is the distance along the rod, t is the time, and I is the moment of inertia per unit length.

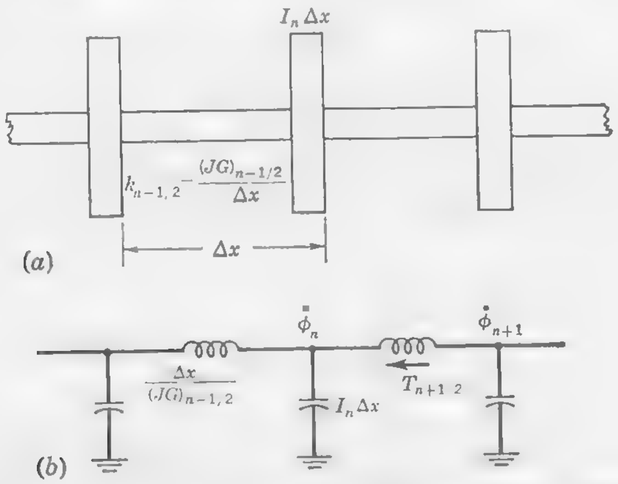


FIG. 9.1.8. The lumped mechanical approximation (*a*) and its exact electrical analogy (*b*) for a torsion rod. Uniform cell size Δx is assumed.

Equation (31) can be replaced by two first-order equations in x, namely,

$$JG\frac{\partial\phi}{\partial x} = T \tag{9.1.32}$$

and

$$\frac{\partial T}{\partial x} = I\frac{\partial^2\phi}{\partial t^2} \tag{9.1.33}$$

where T is the twisting moment. The torsion rod need not be uniform in this case. The finite-difference approximations for Eqs. (32) and (33) are

$$T_{n+\frac12} = (JG)_{n+\frac12}\frac{\phi_{n+1}-\phi_n}{\Delta x} \tag{9.1.34}$$

$$\frac{T_{n+\frac12}-T_{n-\frac12}}{\Delta x} = I_n\frac{d^2\phi_n}{dt^2} \tag{9.1.35}$$

where Δx is the cell length. These finite-difference equations represent *exactly* the lumped mechanical system shown in Fig. 9.1.8*a*. The *exact* nodal analogy for the lumped mechanical system is shown in Fig. 9.1.8*b*, assuming unity scale factors. Note the topological similarity of the two systems.

9.1.8. Beams in Bending. The partial differential equation of motion for a beam in bending is

$$\frac{\partial^2}{\partial x^2}\left(EI\frac{\partial^2 h}{\partial x^2}\right) = -\rho\frac{\partial^2 h}{\partial t^2} \tag{9.1.36}$$

where h is the beam deflection, x is the distance along the rod, t is the time, EI is the bending stiffness, and ρ is the beam mass per unit length. The beam may also be described in terms of four first-order equations

$$\frac{\partial h}{\partial x} = \theta \tag{9.1.37}$$

$$EI\frac{\partial\theta}{\partial x} = M \tag{9.1.38}$$

$$\frac{\partial M}{\partial x} = V \tag{9.1.39}$$

$$\frac{\partial V}{\partial x} = -\rho\frac{\partial^2 h}{\partial t^2} \tag{9.1.40}$$

where θ is the slope of the beam, M is the bending moment, and V is the shear force.

The finite-difference approximations for Eqs. (37) through (40) are

$$\frac{h_{n+1}-h_n}{\Delta x} = \theta_{n+\frac12} \tag{9.1.41}$$

$$(EI)_n\frac{\theta_{n+\frac12}-\theta_{n-\frac12}}{\Delta x} = M_n \tag{9.1.42}$$

$$\frac{M_{n+1}-M_n}{\Delta x} = V_{n+\frac12} \tag{9.1.43}$$

$$\frac{V_{n+\frac12}-V_{n-\frac12}}{\Delta x} = -\rho_n\frac{d^2h_n}{dt^2} \tag{9.1.44}$$

These equations represent the lumped mechanical system of Fig. 9.1.9*a* *exactly* for small deflections. The levers are assumed to be rigid and massless with pin joints at the junctions. The springs between the levers represent lumped bending stiffness and are unstressed when $h = 0$ at all stations.

The analogous electric circuit is shown in Fig. 9.1.9*b*.[1,10] This circuit can be shown to be described by Eqs. (41) through (44). [Actually, Eqs. (41) and (42) are differentiated with respect to time.]

The actual values of the circuit elements can be calculated using Eqs. (10) and (12). It is most convenient to use different voltage scale factors for the h and θ coordinates. Let

$$a_h = Pa_\theta \tag{9.1.45}$$

This implies an increase by a factor P in the relative magnitude of the θ voltages compared with the h voltages. This also means that the transformer turns ratios are $P/\Delta x$ rather than $1/\Delta x$. The constant P is usually chosen to give transformer turns ratios near unity. For the case where the cell length Δx is not a constant for different stations, P is chosen to be near the average cell length.

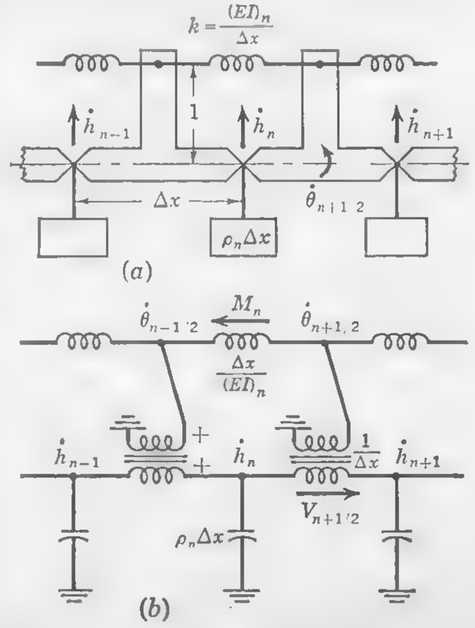


FIG. 9.1.9. (*a*) Mechanical equivalent of finite-difference representation of beam in bending. (*b*) The nodal electrical analogy corresponding to (*a*).

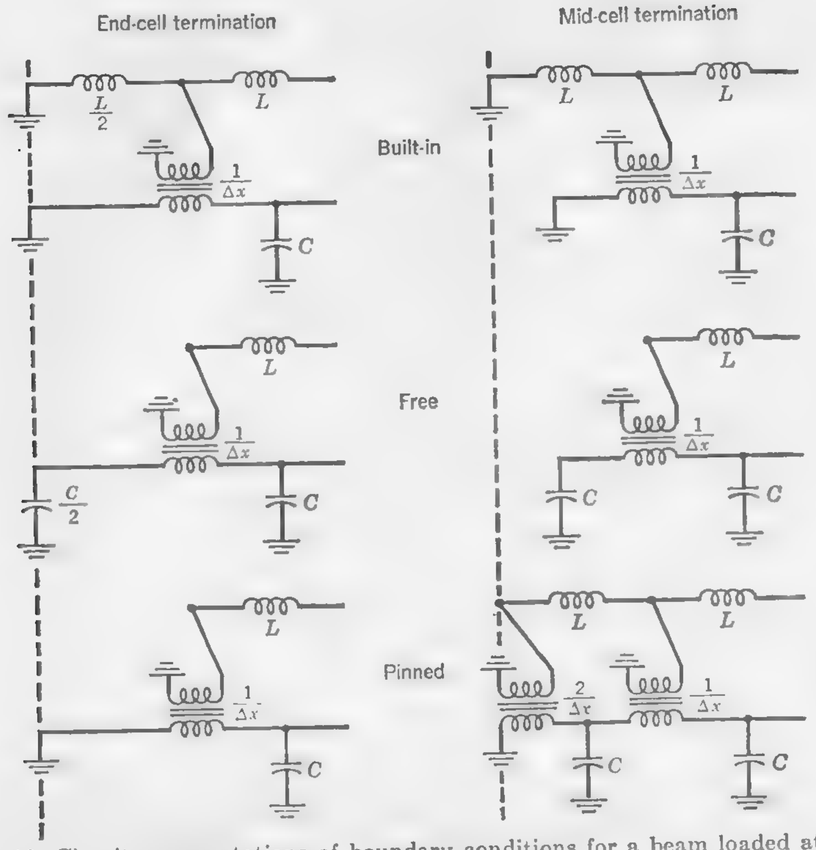


FIG. 9.1.10. Circuit representations of boundary conditions for a beam loaded at the cell ends. The dashed line indicates the end of the beam.

Table 9.1.2. Finite-difference Approximations for Beam Boundary Conditions

Type of boundary condition	Station and end of beam at boundary			
	n		$n + \frac{1}{2}$	
	Left	Right	Left	Right
Free	$M_n = 0$ $V_{n+\frac{1}{2}} = -V_{n-\frac{1}{2}}$	$M_n = 0$ $V_{n-\frac{1}{2}} = -V_{n+\frac{1}{2}}$	$M_{n+1} = M_n = 0$ $V_{n+\frac{1}{2}} = 0$	$M_n = M_{n+1} = 0$ $V_{n+\frac{1}{2}} = 0$
Built-in	$h_n = 0$ $\theta_{n+\frac{1}{2}} = -\theta_{n-\frac{1}{2}}$	$h_n = 0$ $\theta_{n-\frac{1}{2}} = -\theta_{n+\frac{1}{2}}$	$h_{n+1} = h_n = 0$ $\theta_{n+\frac{1}{2}} = 0$	$h_n = h_{n+1} = 0$ $\theta_{n+\frac{1}{2}} = 0$
Pinned	$h_n = 0$ $M_n = 0$	$h_n = 0$ $M_n = 0$	$h_{n+1} = -h_n$ $M_{n+1} = -M_n$	$h_n = -h_{n+1}$ $M_n = -M_{n+1}$

NOTE: Assume a virtual beam extending beyond the boundary in applying the above conditions.

9.1.9. Beam Boundary Conditions. The representation of boundary conditions will be considered for case of free, built-in, and pinned ends where the termination can occur at either the middle of the cell or at the cell end.

At a free end
$$V = 0 \quad \text{and} \quad M = 0 \tag{9.1.46}$$
At a built-in end
$$h = 0 \quad \text{and} \quad \theta = 0 \tag{9.1.47}$$
At a pinned end
$$h = 0 \quad \text{and} \quad M = 0 \tag{9.1.48}$$

The finite-difference approximations for these boundary conditions are shown in Table 9.1.2. The corresponding circuit conditions are shown in Fig. 9.1.10. The mid-cell termination is used more frequently because of better accuracy.

9.1.10. Beams with Rotary Inertia. The beam representation that we have considered so far has not specifically included rotary inertia. However, the beam circuit can be changed relatively easily to include rotary-inertia effects. First, change the location of the capacitors in the h circuit to the transformer center taps. This corresponds to lumping the mass at mid-cell (the center of mass of the cell) instead of at the cell ends and removes the rotary inertia of each cell. The actual rotary inertia can now be added independently by adding a corresponding capacitor from each θ node to ground as shown in Fig. 9.1.11.

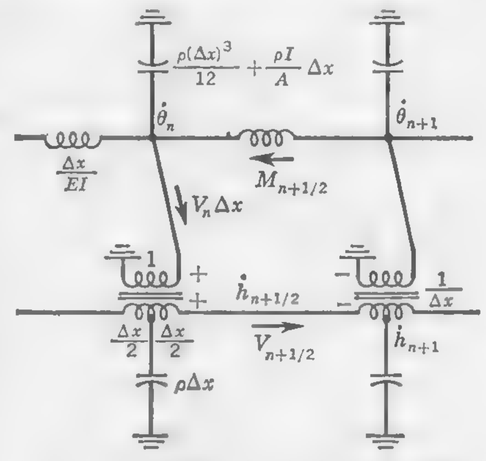


FIG. 9.1.11. Beam circuit for the case of mass lumping at mid-cell and with rotary inertia.

The corresponding finite-difference equations for the lumped mechanical system are

$$\frac{\dot{h}_{n+\frac{1}{2}} - \dot{h}_{n-\frac{1}{2}}}{\Delta x} = \theta_n \tag{9.1.49}$$

$$(EI)_{n+\frac{1}{2}} \frac{\theta_{n+1} - \theta_n}{\Delta x} = \dot{M}_{n+\frac{1}{2}} \tag{9.1.50}$$

$$M_{n+\frac{1}{2}} - M_{n-\frac{1}{2}} = V_n \Delta x + \left(\frac{\rho \Delta x^3}{12} + \rho \frac{I}{A} \Delta x\right) \frac{d^2\theta_n}{dt^2} \tag{9.1.51}$$

$$V_{n+\frac{1}{2}} - V_{n-\frac{1}{2}} = \rho \Delta x \frac{d^2 h_n}{dt^2} \tag{9.1.52}$$

where we have taken

$$\dot{h}_n = \tfrac{1}{2}(\dot{h}_{n+\frac{1}{2}} + \dot{h}_{n-\frac{1}{2}}) \tag{9.1.53}$$

and

$$V_n = \tfrac{1}{2}(V_{n+\frac{1}{2}} + V_{n-\frac{1}{2}}) \tag{9.1.54}$$

The circuit has been shown for the case of a uniform beam. If the beam is not uniform, lump the mass of each cell at the center of mass. This means that the capacitors in the h circuit would be connected to a transformer tap that is not necessarily at the center of the winding. The capacitors in the θ circuit would each correspond to the total moment of inertia of the given cell about its center of mass.

Boundary conditions are applied to this beam representation as shown by the circuits of Fig. 9.1.12.

9.1.11. The Russell Beam Analogy.[12,13] One method of checking the accuracy of representation of the elastic properties of beams is by comparing influence coefficients, i.e., the deflections per unit load. Let us do this for a cantilever beam represented by a

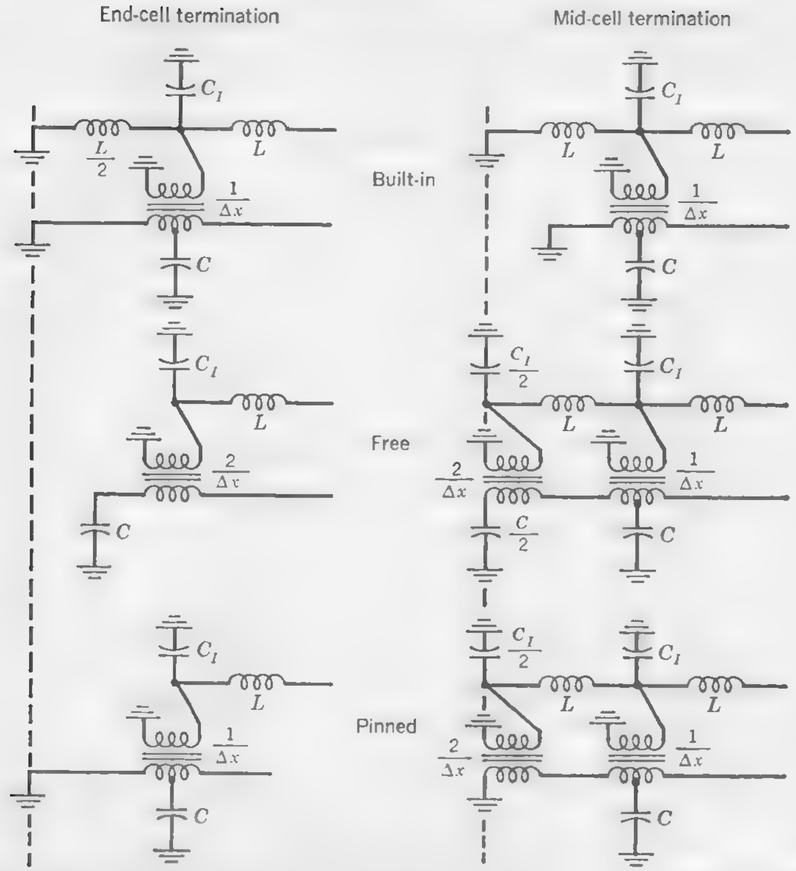


FIG. 9.1.12. Circuit representation of boundary conditions for a beam with mid-cell loading and rotary inertia. The dashed line indicates the end of the beam.

single cell (with mid-cell terminations) and loaded at the end, as shown in Fig. 9.1.13. This system will have a slope and deflection at the end given by

$$\theta = \frac{M\,\Delta x}{EI} + \frac{P(\Delta x)^2}{2EI} \tag{9.1.55}$$

$$h = \frac{M(\Delta x)^2}{2EI} + \frac{P(\Delta x)^3}{4EI} \tag{9.1.56}$$

Now compare these values with those for an actual cantilever beam of length Δx.

$$\theta = \frac{M\,\Delta x}{EI} + \frac{P(\Delta x)^2}{2EI} \tag{9.1.57}$$

$$h = \frac{M(\Delta x)^2}{2EI} + \frac{P(\Delta x)^3}{3EI} \tag{9.1.58}$$

One can see that the slopes agree exactly but the lumped system has a deflection due to P that is too small by an amount $P(\Delta x)^3/12EI$. This error can be corrected by substituting for the pin joint a shear spring of stiffness $12EI/(\Delta x)^3$.

Now an *exact* mechanical model has been obtained for a beam cell which is loaded at the cell ends (where slope and deflection are also measured). This system is the mechanical equivalent of the Russell beam analogy.

Even though the above discussion referred to a uniform beam, the Russell beam analogy can be used to obtain an exact elastic description of a beam cell with nonuniform stiffness, provided that the loading occurs only at the cell ends. In this more general case the shear spring is placed at the centroid of the area under the $1/EI$ curve for the cell. The reciprocal of the stiffness of the shear spring is equal to the moment of inertia of the $1/EI$ curve about its centroid.

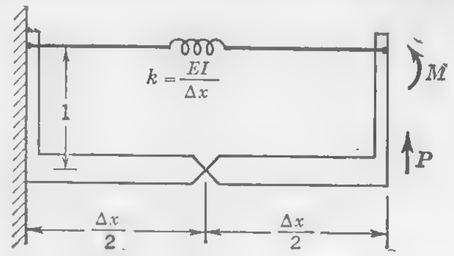


FIG. 9.1.13. A one-cell representation of a uniform cantilever beam.

The lumped mechanical system and its electrical equivalent are shown in Fig. 9.1.14 for the case of an arbitrary stiffness distribution. The electric circuit requires two inductors, one capacitor, and one transformer per cell for the general interior cell. One extra transformer is required at the end. Note that, when two cells are joined, the two transformers at the junction can be replaced by a single transformer with a tapped winding. The capacitors representing the mass loading are connected to the transformer taps.

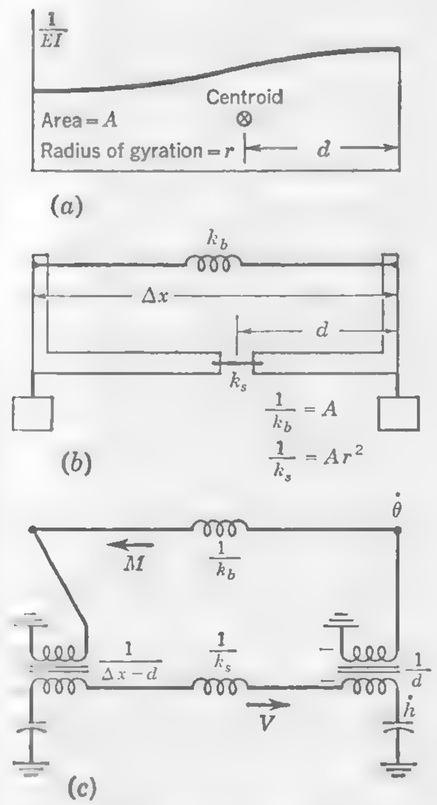


FIG. 9.1.14. The Russell beam analogy for one cell. (*a*) The $1/EI$ curve. (*b*) The lumped mechanical system. (*c*) The electric circuit.

9.1.12. Coupled Bending and Torsion of a Beam. Circuits have been obtained previously for representing a beam in bending or in torsion, but thus far we have not considered

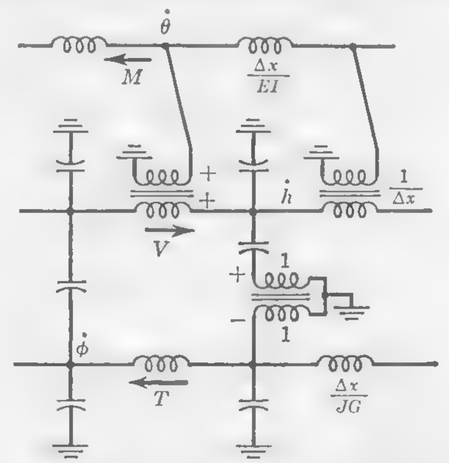


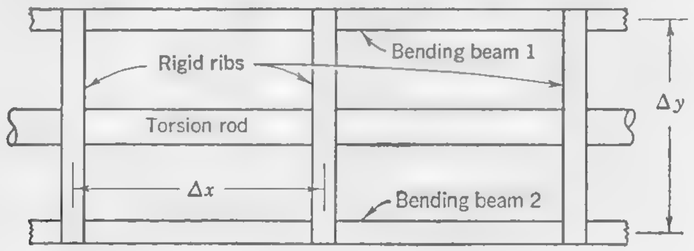
FIG. 9.1.15. The circuit representing coupled bending and torsion of a beam.

simultaneous bending and torsion. The circuit to be presented here assumes that deflections are measured along a straight elastic axis and no elastic coupling exists. The coupling is entirely inertial. This implies that the bending and torsion circuits will be coupled by capacitors representing lumped mutual mass terms (see Fig. 9.1.15).

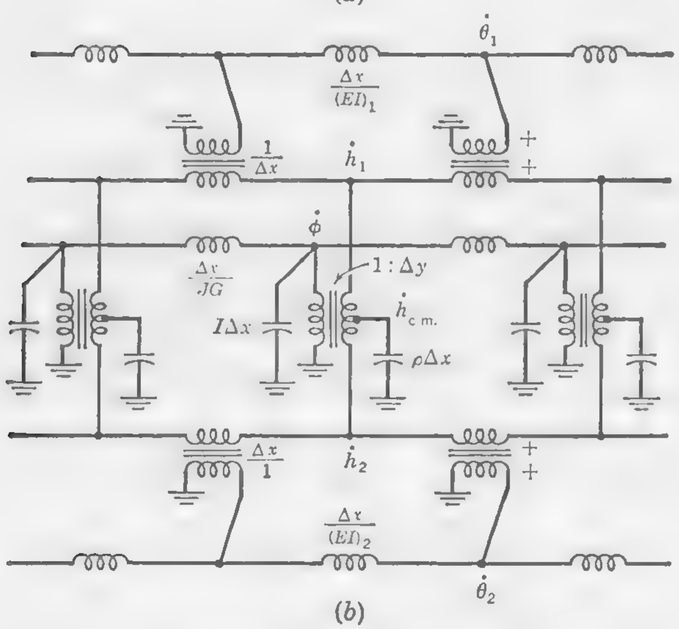
Usually the scale factors in the bending and torsion circuits will be different. This should cause no difficulty if the standard scaling formulas are used and if the actual capacitor sizes are obtained in a manner similar to the example of Fig. 9.1.2.

The beam circuit shown in Fig. 9.1.15 is the basic bending circuit with masses lumped at the cell ends. Other bending circuits could also be used. The torsion circuit toward the bottom of the figure is the standard circuit of Fig. 9.1.8. Note that one negative and one positive mutual mass are assumed.

9.1.13. The Three-beam Circuit. The three-beam circuit provides another means of representing a relatively long and narrow structure such as an aircraft wing of high aspect ratio. It gives a much better structural representation than the simple bending-torsion circuit of Fig. 9.1.15 without undue added complication.



(*a*)



(*b*)

FIG. 9.1.16. The three-beam circuit. (*a*) The mechanical system. (*b*) The electric circuit for uniform characteristics.

The mechanical system is shown in Fig. 9.1.16*a*. There are two beams with stiffness in bending only and a torsion rod with stiffness in torsion only. These beams are connected together by transverse rigid ribs. Note that the structure gets only part of its torsional stiffness from the torsion rod. The remainder comes from the *differential bending* of the two bending beams.

To illustrate the differential bending effect, suppose that the structure is built in at one end and equal and opposite transverse forces are applied at the ends of a rib. This will cause the two bending beams to deflect in opposite directions, giving the effect of a twist to a structure. But the applied forces constitute a couple and therefore a torsional stiffness could be defined at that point. One difference in the nature of the deflection compared with a simple torsion rod is that in this case some additional twisting occurs even outboard of the applied moment.

The three-beam circuit is shown in Fig. 9.1.16*b*. The elastic properties of the torsion rod and bending beams are represented in a manner identical to that given previously.

Each rib is considered to be rigid and is represented by a transformer winding. The other winding in each case generates the ϕ for that station and thereby connects with the torsion circuit. The capacitor representing the mass of each cell is connected to a transformer tap corresponding to the center-of-mass location for that cell. The capacitors in the torsion circuit represent the moment of inertia of each cell about its center of mass. No other capacitors are required for coupling in this case since the transformers provide the necessary coupling.

A comparison of the simple bending-torsion circuit with the three-beam circuit shows that the three-beam circuit requires three inductors, two capacitors, and three transformers per cell, whereas the simple bending-torsion circuit requires two inductors, three capacitors, and an average of about 1½ transformers per cell.

The three-beam circuit has been shown for the case of a uniform structure; i.e., each cell is similar to the next. If the structure is not uniform, the individual elements are adjusted accordingly. For example, a beam with relatively small taper can be adequately represented by changing the coupling transformer turns ratio according to the local Δy.

If the elastic properties of the structure are given in terms of influence coefficients, a trial-and-error method using the computer can often be resorted to in order to determine proper values of the circuit elements. In this case, a sinusoidal forcing function is used whose frequency is of the order of the important solution frequencies. *All capacitors are disconnected when matching influence coefficients.*

ANALOGIES FOR STATIC STRUCTURES

9.1.14. A Comparison of Static and Dynamic Circuits. The analogies which have been presented so far are quite convenient for making transient analyses. If properly scaled, they will represent satisfactorily the lumped mechanical system over its entire range of natural frequencies, with the exception of any zero-frequency modes.

For the case where static analyses are to be made, the conversion from the dynamic circuit to the corresponding static circuit is straightforward. The procedure is as follows:

1. Remove all inductors and capacitors from the dynamic circuit.
2. Replace all inductors by resistors of the same nominal size. (Of course, in each case, the actual inductor or resistor size is determined by using the scaling equations with the appropriate scale factors.)
3. Transformer turns ratios and tap connections remain unchanged.

Static structural circuits should be driven with a sinusoidal current or voltage at a frequency such that all elements are operating well within their normal range. Measurements are made by metering a-c voltages and currents corresponding to d-c displacements and forces.

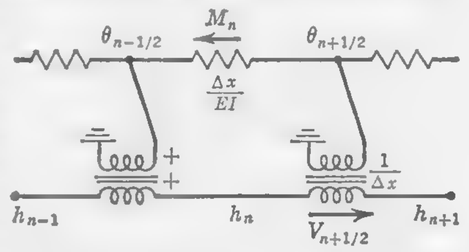


FIG. 9.1.17. The static structural circuit for a simple finite-difference beam.

The correspondence between a-c quantities in the electric circuit and d-c quantities in the mechanical system is valid because inertial effects have been removed with the removal of the capacitors. Therefore the ratio of a-c forces and displacements is the same as for d-c forces and displacements.

The replacement of inductors by resistors is principally a matter of convenience. Resistors cost less than inductors and their electrical characteristics are less sensitive to changes in frequency or power level.

The currents in the transformer-resistor circuit are 90 deg leading in phase compared with the corresponding currents in the transformer-inductor circuit. However, the magnitude ratios in each case are the same. In general, if the elements, exclusive of transformers, are all of the same type and have corresponding impedances, then the current and voltage ratios will be independent of the element type.

As a comparison of static and dynamic circuits, note the static-beam circuit of Fig. 9.1.17 and the corresponding dynamic circuit previously shown in Fig. 9.1.9*b*.

9.1.15. The Plane-stress Circuit. The plane-stress problem is essentially a two-dimensional problem. An example might be the analysis of stresses and strains in a thin sheet loaded by forces in its plane. These forces will produce both normal and shear stresses and strains.

The stress-strain relationships are

$$e_x = \frac{1}{E}(s_x - \mu s_y) \tag{9.1.59}$$

$$e_y = \frac{1}{E}(s_y - \mu s_x) \tag{9.1.60}$$

$$\gamma = \frac{1}{G} s_s \tag{9.1.61}$$

where E = Young's modulus

$G = \dfrac{E}{2(1+\mu)}$ = shear modulus

s_x, s_y = normal stresses on planes perpendicular to the x and y axes, respectively

s_s = shear stress

e_x, e_y = normal strains in x and y directions, respectively

γ = shear strain

μ = Poisson's ratio

The compatibility equation in terms of stress components is

$$\frac{\partial^2 s_x}{\partial x^2} + \frac{\partial^2 s_y}{\partial y^2} + 2\frac{\partial^2 s_s}{\partial x \partial y} = 0 \tag{9.1.62}$$

Let u and v be the displacements in the x and y directions, respectively. The cell dimensions are Δx by Δy. Assume a unit thickness throughout. Then Eqs. (59) to (62) can be represented in finite-difference form by the circuit of Fig. 9.1.18.

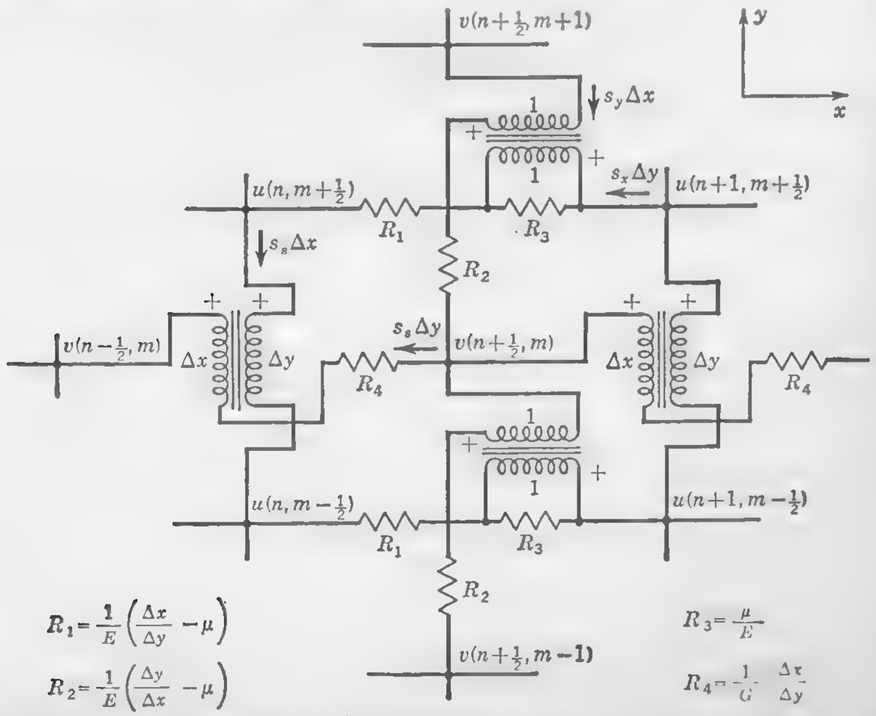


FIG. 9.1.18. The plane-stress circuit. Displacements in the x direction are denoted by u, displacements in the y direction by v. Unit thickness is assumed. For other thicknesses t, divide all resistors by t.

From the circuit one sees that

$$e_x \Delta x = u\left(n+1, m+\frac{1}{2}\right) - u\left(n, m+\frac{1}{2}\right)$$

$$= (s_x \Delta y - s_y \Delta x)R_3 + s_x \Delta y R_1 = \frac{\Delta x}{E}(s_x - \mu s_y)$$

in agreement with Eq. (59). Similarly,

$$e_y \Delta y = v\left(n+\frac{1}{2}, m+1\right) - v\left(n+\frac{1}{2}, m\right)$$

$$= \frac{s_y \Delta x}{E}\left(\frac{\Delta y}{\Delta x} - \mu\right) - \frac{\mu}{E}(s_x \Delta y - s_y \Delta x) = \frac{\Delta y}{E}(s_y - \mu s_x)$$

in agreement with Eq. (60).

The expression for the shear strain γ can be obtained by noting

$$\gamma = \frac{u(n, m+\frac{1}{2}) - u(n, m-\frac{1}{2})}{\Delta y} + \frac{v(n+\frac{1}{2}, m) - v(n-\frac{1}{2}, m)}{\Delta x} \tag{9.1.63}$$

But from the circuit

$$v\left(n+\frac{1}{2}, m\right) - v\left(n-\frac{1}{2}, m\right) = -\left[u\left(n, m+\frac{1}{2}\right) - u\left(n, m-\frac{1}{2}\right)\right]\frac{\Delta x}{\Delta y} + s_s \Delta y R_4$$

giving

$$\gamma = s_s \frac{\Delta y}{\Delta x} R_4 = \frac{s_s}{G}$$

in agreement with Eq. (61).

There remains a check of the compatibility equation. First, sum currents at the node $v(n+\frac{1}{2}, m)$:

$$s_y \Delta x(n+\tfrac{1}{2}, m+\tfrac{1}{2}) - s_y \Delta x(n+\tfrac{1}{2}, m-\tfrac{1}{2}) + s_s \Delta y(n+1, m) - s_s \Delta y(n, m) = 0$$

This yields the finite-difference form of the equation

$$\frac{\partial s_y}{\partial y}\left(n+\frac{1}{2}, m\right) + \frac{\partial s_s}{\partial x}\left(n+\frac{1}{2}, m\right) = 0 \tag{9.1.64}$$

after dividing by $\Delta x \, \Delta y$. Similarly,

$$\frac{\partial s_y}{\partial y}\left(n+\frac{1}{2}, m+1\right) + \frac{\partial s_s}{\partial x}\left(n+\frac{1}{2}, m+1\right) = 0 \tag{9.1.65}$$

Using the same approach we can also obtain

$$\frac{\partial s_x}{\partial x}\left(n, m+\frac{1}{2}\right) + \frac{\partial s_s}{\partial y}\left(n, m+\frac{1}{2}\right) = 0 \tag{9.1.66}$$

and

$$\frac{\partial s_x}{\partial x}\left(n+1, m+\frac{1}{2}\right) + \frac{\partial s_s}{\partial y}\left(n+1, m+\frac{1}{2}\right) = 0 \tag{9.1.67}$$

Now subtract Eq. (64) from (65), obtaining the finite-difference form of

$$\frac{\partial^2 s_y}{\partial y^2}\left(n+\frac{1}{2}, m+\frac{1}{2}\right) + \frac{\partial^2 s_s}{\partial x \, \partial y}\left(n+\frac{1}{2}, m+\frac{1}{2}\right) = 0 \tag{9.1.68}$$

after dividing by Δy. Similarly, by subtracting Eq. (66) from (67) we can obtain the finite-difference form of

$$\frac{\partial^2 s_x}{\partial x^2}\left(n+\frac{1}{2}, m+\frac{1}{2}\right)+\frac{\partial^2 s_s}{\partial x\,\partial y}\left(n+\frac{1}{2}, m+\frac{1}{2}\right)=0 \tag{9.1.69}$$

Finally, adding Eqs. (68) and (69) we obtain

$$\frac{\partial^2 s_x}{\partial x^2}\left(n+\frac{1}{2}, m+\frac{1}{2}\right)+2\frac{\partial^2 s_s}{\partial x\,\partial y}\left(n+\frac{1}{2}, m+\frac{1}{2}\right)+\frac{\partial^2 s_y}{\partial y^2}\left(n+\frac{1}{2}, m+\frac{1}{2}\right)=0 \tag{9.1.70}$$

which is the compatibility equation for the point $(n + \frac{1}{2}, m + \frac{1}{2})$.

Therefore, Fig. 9.1.18 does represent in finite-difference form the equations of plane stress.

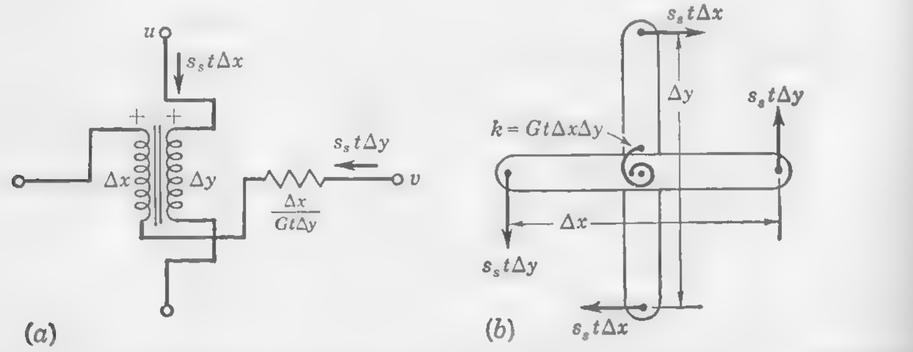


FIG. 9.1.19. (*a*) The shear-stress circuit and (*b*) its mechanical analogy. The levers are assumed to be rigid. The thickness is t and the torsional spring constant is k.

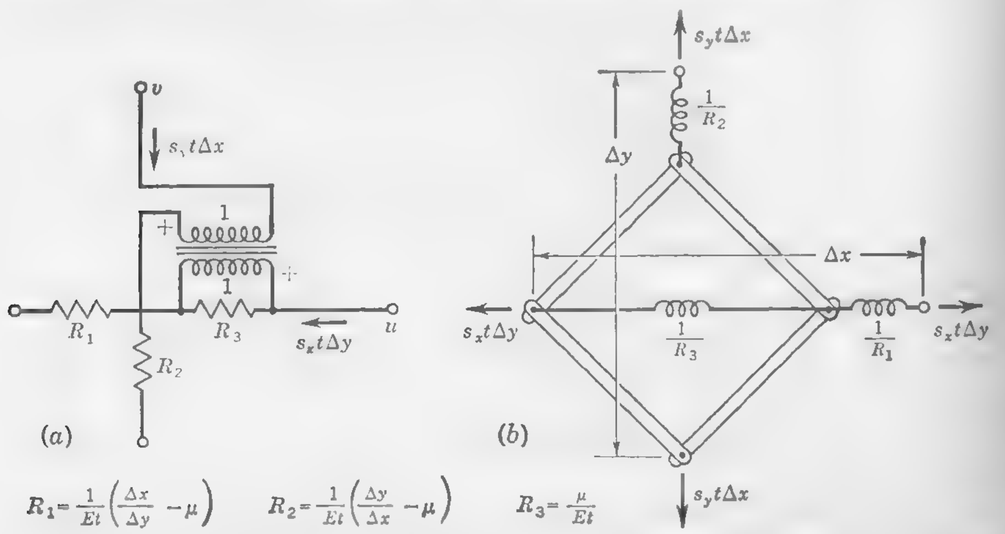


FIG. 9.1.20. (*a*) The normal-stress circuit. (*b*) The mechanical analogy. The levers are equal in length. Small displacements are assumed.

It can be seen from the circuit that it is composed of two types of simpler circuits connected in an alternating or checkerboard fashion. These two circuits represent separately the shear stresses and strains and the normal stresses and strains. It is possible to obtain a mechanical analogy for each of these simpler circuits and thereby gain further insight into the problem.

The shear-stress circuit and its mechanical analogy are shown in Fig. 9.1.19. The normal-stress circuit and the mechanical analogy are shown in Fig. 9.1.20.

9.1.16. The Plate Analogy.[6] Small deflections of a laterally loaded plate may be calculated from the following six equations:

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2}+\mu\frac{\partial^2 w}{\partial y^2}\right) \tag{9.1.71}$$

$$M_y = -D\left(\frac{\partial^2 w}{\partial y^2}+\mu\frac{\partial^2 w}{\partial x^2}\right) \tag{9.1.72}$$

$$M_{yx} = -M_{xy} = -D(1-\mu)\frac{\partial^2 w}{\partial x\,\partial y} \tag{9.1.73}$$

$$Q_x = \frac{\partial M_x}{\partial x}+\frac{\partial M_{yx}}{\partial y} \tag{9.1.74}$$

$$Q_y = \frac{\partial M_y}{\partial y}+\frac{\partial M_{yx}}{\partial x} \tag{9.1.75}$$

$$\frac{\partial Q_x}{\partial x}+\frac{\partial Q_y}{\partial y}+q=0 \tag{9.1.76}$$

where $D = \dfrac{Et^3}{12(1-\mu^2)}$

w = deflection in the z direction
M = bending or twisting moments per unit edge length (see Fig. 9.1.21)
Q = shear force per unit edge length
q = load per unit area

Using the same procedures that have been applied previously, we can show that the circuit of Fig. 9.1.22 represents the finite-difference approximation of Eqs. (71) to (76). Note that this circuit requires three transformers and five resistors per cell.

The case of a plate of varying thickness can be analyzed by calculating local values of the flexural rigidity D and evaluating the resistors accordingly. Resistors R_1, R_2, and R_3 are evaluated at integer stations in x and y, whereas resistors R_4 and R_5 are evaluated at half-integer stations.

Another circuit has been developed by Benscoter and MacNeal[2] for the case of a built-up structure that approximates a plate in its elastic properties. This circuit (Fig. 9.1.24) can be obtained by considering a mechanical model consisting of a rectangular grid of beams covered by a skin on each side (see Fig. 9.1.23).

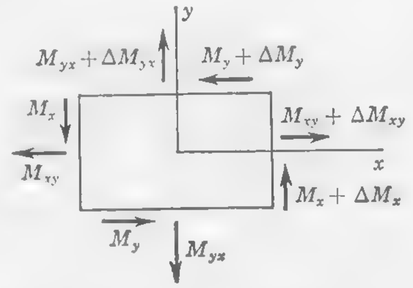
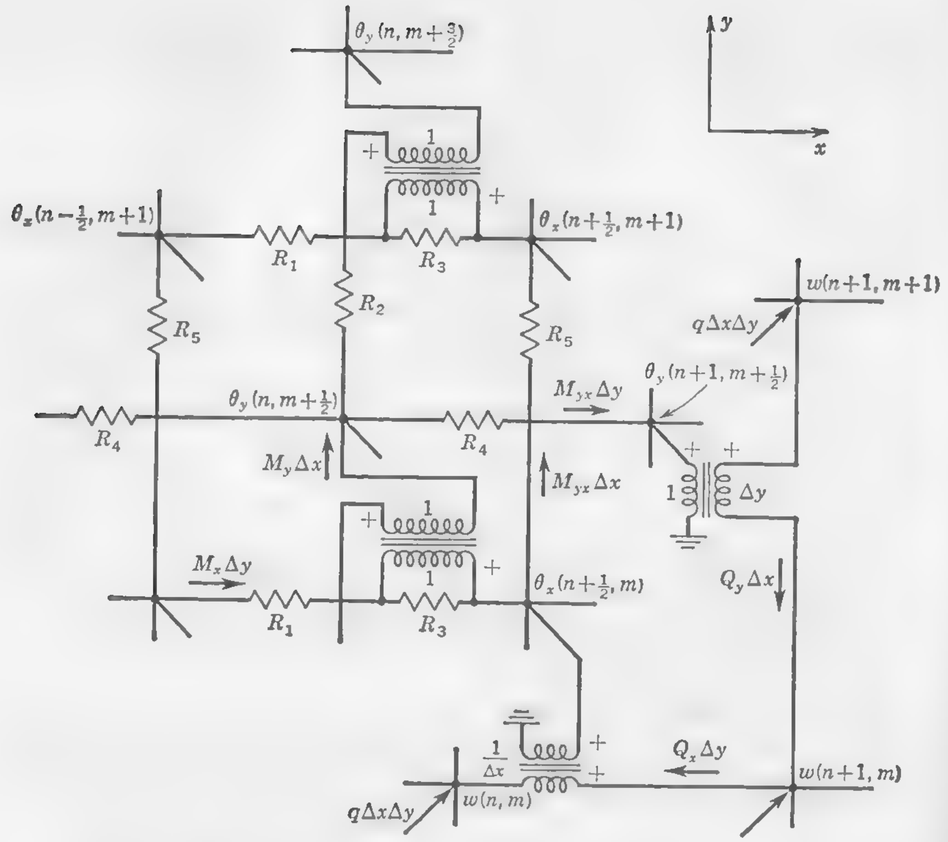


FIG. 9.1.21. The designation of bending and twisting moments acting on a rectangular section of a plate. (Use right-hand rule convention.)

The loads are applied and deflections measured at the beam intersections, i.e., at the cell ends. Displacements of the skin are assumed to be in its plane and are measured at the mid-points of the beam cells. In fact, the circuit representing the skin is identical in form with the plane-stress circuit of Fig. 9.1.18. It is from the plane-stress circuit that the beam bending stiffness is derived. Physically, however, part of the bending stiffness comes from the skin, and part comes from the beam webs. As can be seen from Fig. 9.1.23, displacements of the skin are proportional to β, the angular displacement of the vertical member.

The beams are assumed to deflect because of shear forces as well as bending moments. The shear stiffness is assumed to be entirely in the web and is represented by resistors between the beam and plane-stress circuits.



$$R_1 = \frac{1}{D(1-\mu^2)}\left(\frac{\Delta x}{\Delta y} - \mu\right) \qquad R_3 = \frac{\mu}{D(1-\mu^2)} \qquad R_5 = \frac{\Delta y}{D(1-\mu)\Delta x}$$

$$R_2 = \frac{1}{D(1-\mu^2)}\left(\frac{\Delta y}{\Delta x} - \mu\right) \qquad R_4 = \frac{\Delta x}{D(1-\mu)\Delta y}$$

FIG. 9.1.22. The plate analogy. The plate thickness need not be uniform.

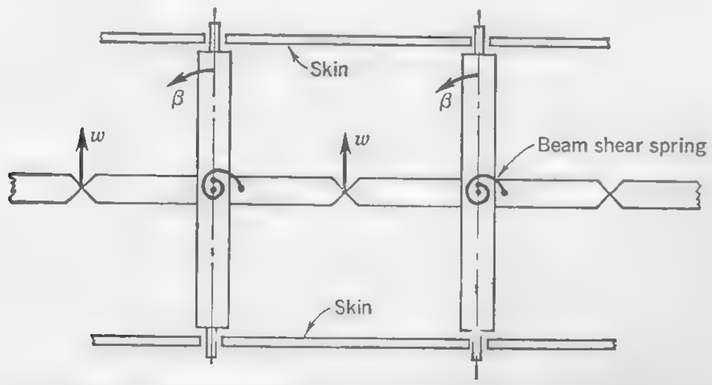
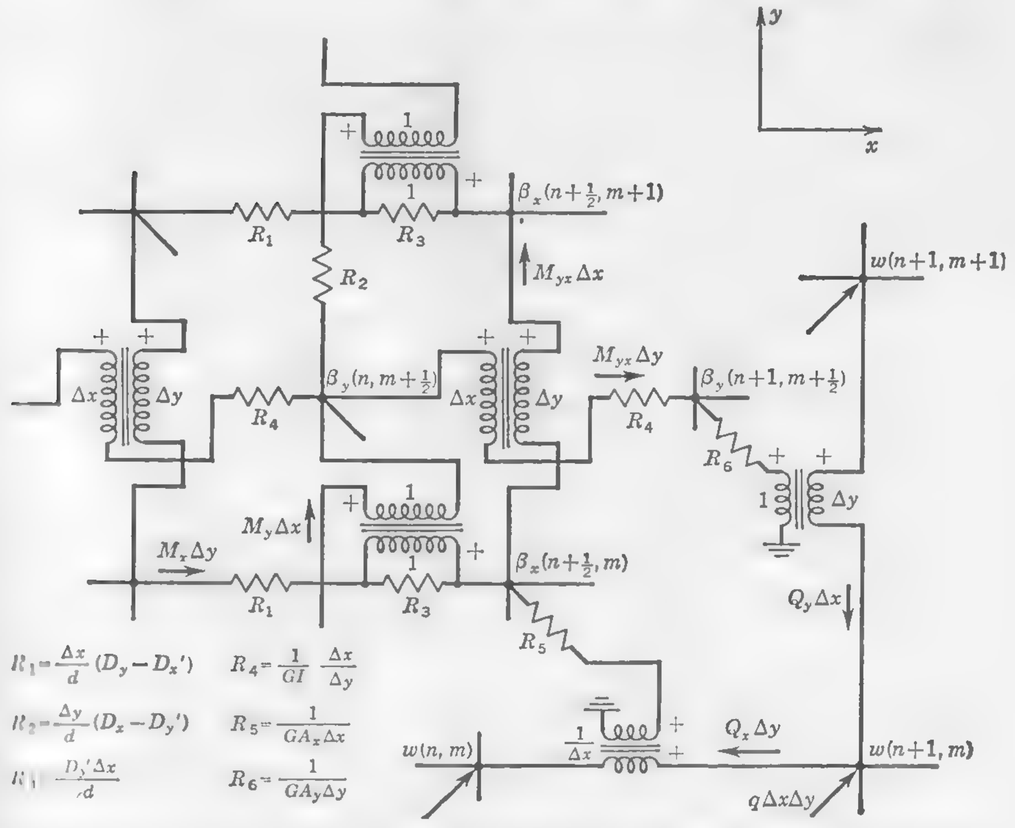


FIG. 9.1.23. Mechanical representation of a built-up plate. Another set of beams is directed out of the page, joining the given beam at the vertical deflection points.



$$R_1 = \frac{\Delta x}{d}(D_y - D_x') \qquad R_4 = \frac{1}{GI}\frac{\Delta x}{\Delta y}$$

$$R_2 = \frac{\Delta y}{d}(D_x - D_y') \qquad R_5 = \frac{1}{GA_x\Delta x}$$

$$R_3 = \frac{D_y'\Delta x}{d} \qquad R_6 = \frac{1}{GA_y\Delta y}$$

FIG. 9.1.24. Circuit for a built-up plate consisting of a rectangular grid of beams and a covering skin.

It can be shown that the structure is represented by the circuit of Fig. 9.1.24, where

$$D_x = \Delta y \frac{EI}{1-\mu^2} + EI_x$$

$$D_y = \Delta x \frac{EI}{1-\mu^2} + EI_y$$

$$D_x' = \Delta y \frac{\mu EI}{1-\mu^2}$$

$$D_y' = \Delta x \frac{\mu EI}{1-\mu^2}$$

$d = D_xD_y - D_x'D_y'$
I = moment of inertia per unit length of the skin cross section about neutral axis of beam
I_x = moment of inertia of web cross section of beam parallel to x axis
I_y = moment of inertia of web cross section of beam parallel to y axis
A_x = area of web cross section of beam parallel to x axis
A_y = area of web cross section of beam parallel to y axis

It can be seen that four transformers and six resistors are required per cell. It can be shown that one transformer and one resistor could be saved per cell if no shear compliance is assumed in the beams. In this case the shear transformer in the plane-stress circuit is superfluous (provided another resistor is used to represent shear compliance in the skin) and the circuit reduces to the usual plate circuit of Fig. 9.1.22.

9.1.17. The Ring Analogy. The importance of circular rings as structural elements in aircraft and missiles need hardly be emphasized. In order to avoid the complexity of representing a ring as a curved beam using coordinate transformations between successive cells, a simpler circuit has been developed by MacNeal.[4]

Consider the lumped mechanical analogy shown in Fig. 9.1.25. The levers are rigid and pinned at the junctions. The coordinate u represents a radial displacement whereas the coordinates β and θ represent angular displacements. External forces may be applied to the system, corresponding to each of the coordinates. Thus N, F, and L are externally applied forces and moments. Figure 9.1.26 shows the various forces and moments that enter into the analysis.

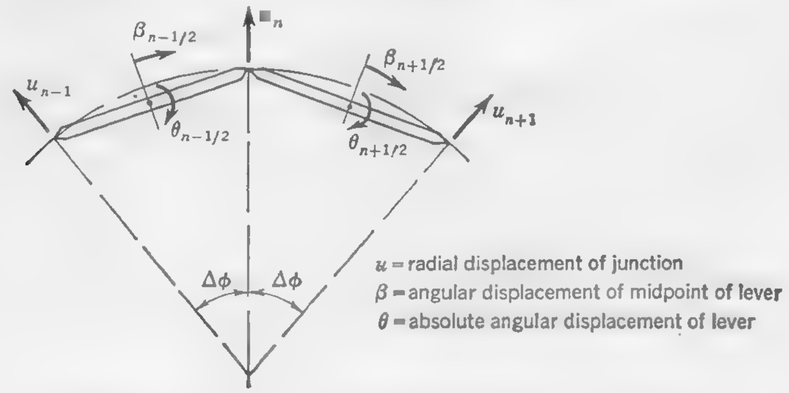


FIG. 9.1.25. The lumped mechanical analogy for a ring.

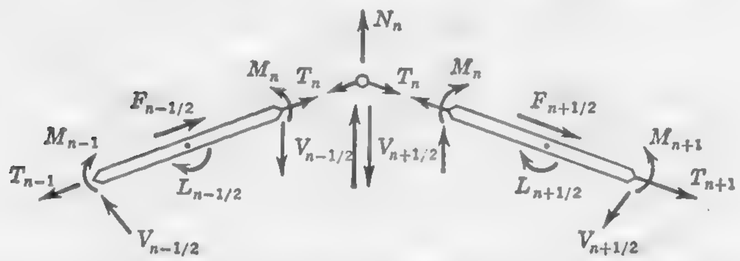


FIG. 9.1.26. An exploded view of mechanical analogy of the ring, showing the forces and moments acting. The pin in the center transmits forces between the levers.

It can be seen that all forces and displacements are in the plane of the ring. Another assumption of the analysis is that of small displacements. With these assumptions, let us check the electrical analogy.

First, notice that a difference in the radial displacement at the two ends of a lever is accompanied by a difference between β and θ.

$$u_n - u_{n+1} = 2r \sin \frac{\Delta\phi}{2} \cos \frac{\Delta\phi}{2} (\theta_{n+\frac{1}{2}} - \beta_{n+\frac{1}{2}})$$

where r is the radius of the ring. This equation may also be written

$$\frac{u_n}{r} - \frac{u_{n+1}}{r} = \sin \Delta\phi(\theta_{n+\frac{1}{2}} - \beta_{n+\frac{1}{2}}) \tag{9.1.77}$$

Because the levers are rigid, the displacement components along the lever must be equal at its two ends.

$$u_n \sin \frac{\Delta\phi}{2} = -r\beta_{n+\frac{1}{2}} \cos \frac{\Delta\phi}{2} \tag{9.1.78}$$

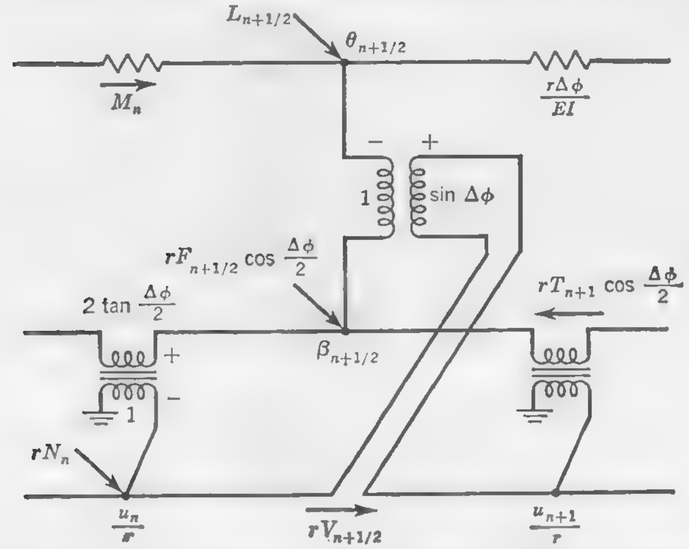


FIG. 9.1.27. The electric circuit representing the bending of a circular ring in its own plane.

Similarly,

$$u_n \sin \frac{\Delta\phi}{2} = r\beta_{n-\frac{1}{2}} \cos \frac{\Delta\phi}{2} \tag{9.1.79}$$

From Eqs. (78) and (79) we obtain

$$\beta_{n+\frac{1}{2}} - \beta_{n-\frac{1}{2}} = -\frac{2u_n}{r} \tan \frac{\Delta\phi}{2} \tag{9.1.80}$$

There remains the specification of the elastic properties of the ring. Assume that the ring is relatively thin compared with the radius of curvature so that ordinary beam theory may be used. We shall lump the bending compliance at the lever junctions. Then

$$\theta_{n-\frac{1}{2}} - \theta_{n+\frac{1}{2}} = \frac{M_n r \Delta\phi}{EI} \tag{9.1.81}$$

From Fig. 9.1.27 it can be seen that Eqs. (77), (80), and (81) are represented by the electric circuit. Two transformers and one resistor are required per cell.

The circuit that we have presented assumes a general loading in the plane of the ring. A special case of particular interest occurs when the ring is a part of a cylindrical shell. In this case the external load comes from the skin and is tangent to the outer surface of the ring. Therefore the radial load N is zero. Furthermore, the tangential force F and the moment L (see Fig. 9.1.26) have a constant ratio which depends upon the eccentricity d of the effective load (see Fig. 9.1.28).

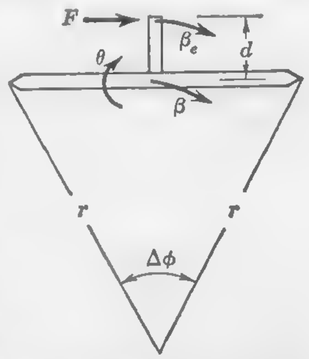


FIG. 9.1.28. The mechanical analogy for a cell of a ring with eccentric tangential loading.

The circuit of Fig. 9.1.27 can be modified slightly to make allowance for the eccentric loading. First, it is necessary to obtain a relationship between β_e and β, where β_e is the angular displacement of the eccentric loading point. The tangential displacements at β and β_e are related by

$$\left(r \cos \frac{\Delta\phi}{2} + d\right) \beta_e = \left(r \cos \frac{\Delta\phi}{2}\right) \beta + \theta d \tag{9.1.82}$$

giving

$$\beta_e = \beta + \frac{d}{d + r\cos(\Delta\phi/2)}(\theta - \beta) \tag{9.1.83}$$

The electrical analogy for the ring with eccentric loading is shown in Fig. 9.1.29.

Electrical analogies can also be constructed for noncircular rings. In this case, the ring is approximated by several circular arcs of differing radii. The circuit is similar to those we have presented except that additional transformers are required in the θ circuit to provide a scale change at points where the radius changes.

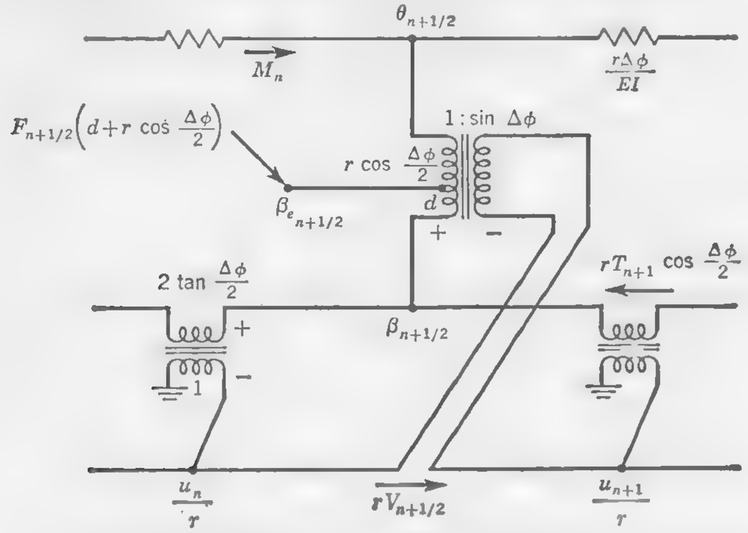


FIG. 9.1.29. The electrical analogy for ring with eccentric loading.

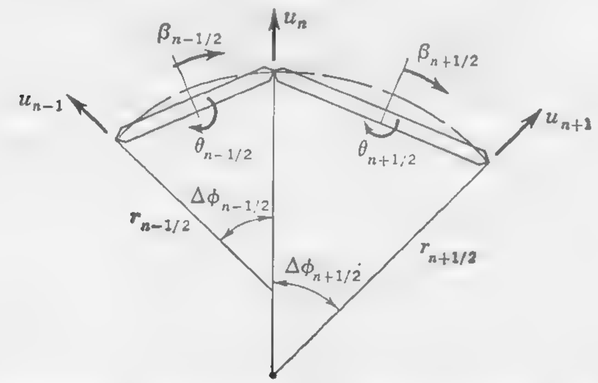


FIG. 9.1.30. The mechanical representation of a noncircular ring with unequal cell size.

The mechanical system for the general case of a noncircular ring with unequal cells is shown in Fig. 9.1.30. The difference equations describing this system are

$$u_n - u_{n+1} = r_{n+\frac{1}{2}}\sin\Delta\phi_{n+\frac{1}{2}}(\theta_{n+\frac{1}{2}} - \beta_{n+\frac{1}{2}}) \tag{9.1.84}$$

$$r_{n+\frac{1}{2}}\beta_{n+\frac{1}{2}} - r_{n-\frac{1}{2}}\beta_{n-\frac{1}{2}} = -u_n\left(\tan\frac{\Delta\phi_{n-\frac{1}{2}}}{2} + \tan\frac{\Delta\phi_{n+\frac{1}{2}}}{2}\right) \tag{9.1.85}$$

$$\theta_{n-\frac{1}{2}} - \theta_{n+\frac{1}{2}} = \frac{M}{2EI}(r_{n-\frac{1}{2}}\Delta\phi_{n-\frac{1}{2}} + r_{n+\frac{1}{2}}\Delta\phi_{n+\frac{1}{2}}) \tag{9.1.86}$$

For the case of unequal cell size the forces and moments are defined as in Fig. 9.1.26 except for the tangential force T. In this case the tangential forces on either side of a

junction are unequal because the angles are unequal. The relation is

$$T_{n-\frac{1}{4}}\cos\frac{\Delta\phi_{n-\frac{1}{2}}}{2} = T_{n+\frac{1}{4}}\cos\frac{\Delta\phi_{n+\frac{1}{2}}}{2} \tag{9.1.87}$$

The electric circuit representing a noncircular ring with unequal cells is shown in Fig. 9.1.31. Comparing this circuit with that for the circular ring (Fig. 9.1.27), it can be seen that the voltage level has been multiplied by the radius of curvature r. In order to represent Eq. (86) properly, one transformer per cell has been added to the θ circuit. Thus the circuit of Fig. 9.1.31 requires three transformers and one resistor per cell. If two interior taps are available on the autotransformer, it can be shown that only one autotransformer is required for every two cells, thereby saving approximately one-half transformer per cell.

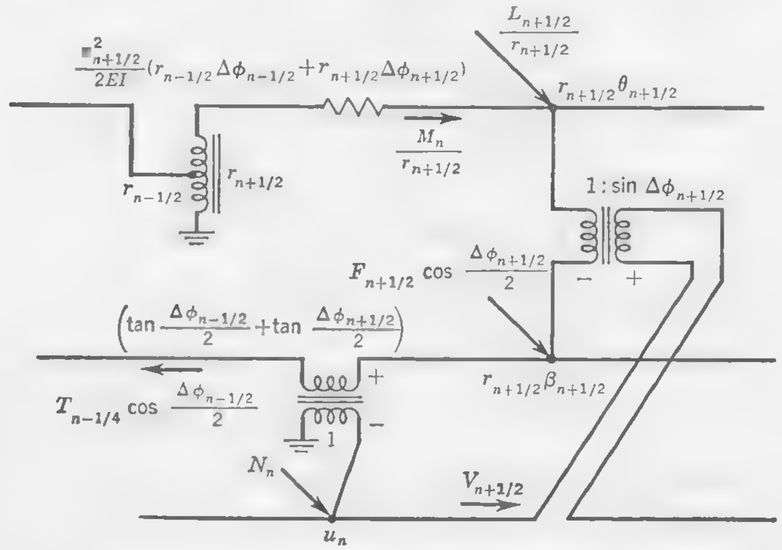


FIG. 9.1.31. The electrical analogy for a noncircular ring with unequal cell size.

9.1.18. The Analogy for a Stiffened Cylindrical Shell. The stiffened cylindrical shell is a common structural component in aircraft and missiles and therefore its static or dynamic characteristics are important in the over-all structural analysis. We shall consider here only circular cylindrical shells, although the same principles can be used to obtain analogies for cylinders with elliptical or other noncircular cross sections.

For the purposes of our analysis, the stiffened cylindrical shell will be assumed to be composed of the following: (1) circular rings, (2) shear panels, and (3) stringers. Each ring is assumed to resist bending in its own plane but to have zero stiffness for deformations perpendicular to its plane. The ring circuit has already been given (Fig. 9.1.27).

Similarly, the shear panel circuit has been shown previously (Fig. 9.1.19). These panels represent the skin and are assumed to be loaded in shear only. The longitudinal forces are transmitted by stringers which are assumed to have zero bending stiffness. Since it transmits only longitudinal forces, each stringer section can be represented by a resistor.

In summary, the ring circuit is not directly connected to the stringer circuit. However, the transverse forces on the rings are transformed to longitudinal forces of stringers by means of the shear panels.

The complete circuit is shown in Fig. 9.1.32 for the case where the skin is eccentrically attached at an effective distance r' from the center of the cylinder, where

$$r' = r\cos\frac{\Delta\phi}{2} + d \tag{9.1.88}$$

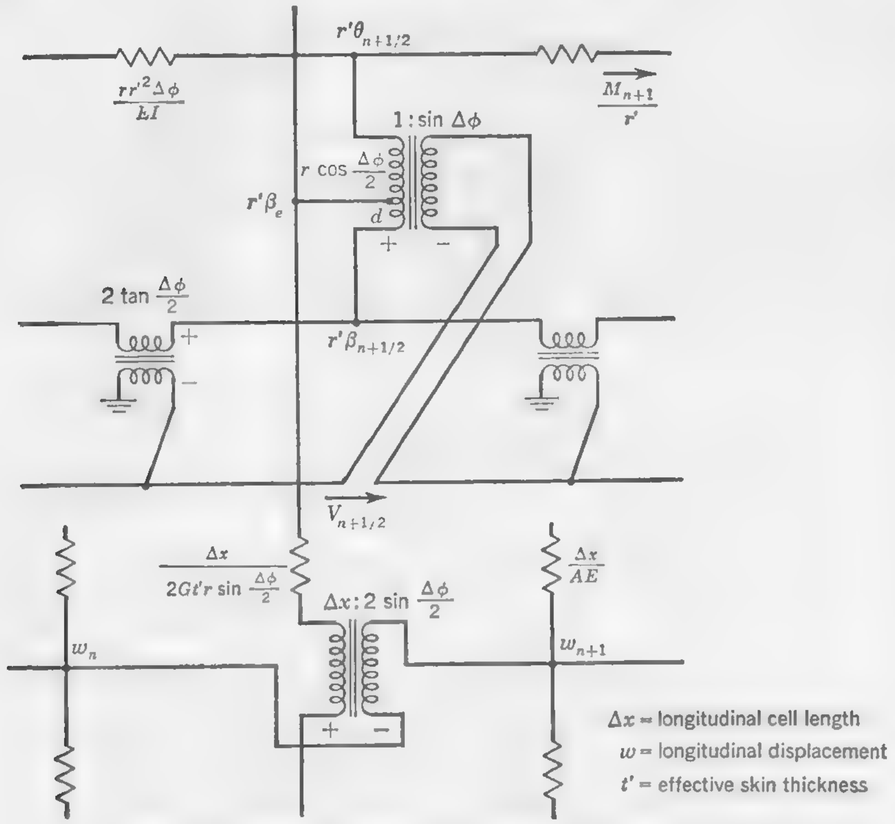


FIG. 9.1.32. The electrical analogy for a stiffened circular cylindrical shell.

Each shear panel is of length Δx, width $2r \sin(\Delta\phi/2)$, and thickness t' where

$$t' = \frac{2 \sin(\Delta\phi/2)}{\Delta\phi} t \tag{9.1.89}$$

and t is the actual skin thickness.

For further details concerning the electrical representation of cylindrical shells the reader is referred to MacNeal's paper.[4]

ERRORS

9.1.19. Types of Errors. The systematic errors that arise in the solutions of problems on the direct-analogy computer are due to (1) finite-difference approximations, (2) calibration errors in computing elements, (3) calibration errors in metering equipment, and (4) parasitic impedances of computing and metering elements.

While calibration errors can be serious, they normally represent small deviations from the intended parameter values for the system. This means that the problem actually put on the computer is somewhat different from the intended problem, but generally the solutions are qualitatively similar. We shall not consider further the calibration errors but shall restrict ourselves to a discussion of errors due to finite-difference approximations and parasitic impedances.

9.1.20. Finite-difference Approximations. We have seen previously that a beam or a plate that is described in terms of partial differential equations is not represented directly on the computer but is approximated by a system describable in terms of a set of ordinary differential equations (or even algebraic equations). This simplification of the problem implies a lumping of the inertial and elastic properties of the

system which can be derived by using finite-difference approximations for the spatial derivatives.

In structural-dynamics problems the finite-difference errors appear as errors in the natural frequencies and mode shapes. These errors are generally more pronounced for the higher modes but should tend toward zero as the number of cells becomes very large. We shall consider in particular the finite-difference errors in the vibration modes of a beam.

First recall the bending equations (37) to (40) and the corresponding central-difference approximations given in Eqs. (41) to (44) for the beam of Fig. 9.1.9. If these equations are combined, central-difference approximations to the second, third, and fourth derivatives can be found. These approximations can also be derived by using a Taylor expansion about the point where the derivative is to be calculated. The results are

$$\left.\frac{\partial h}{\partial x}\right|_{n+\frac{1}{2}} = \frac{h_{n+1} - h_n}{\Delta x} - \frac{(\Delta x)^2}{24} \left.\frac{\partial^3 h}{\partial x^3}\right|_{n+\frac{1}{2}} \tag{9.1.90}$$

$$\left.\frac{\partial^2 h}{\partial x^2}\right|_{n} = \frac{h_{n+1} - 2h_n + h_{n-1}}{(\Delta x)^2} - \frac{(\Delta x)^2}{12} \left.\frac{\partial^4 h}{\partial x^4}\right|_{n} \tag{9.1.91}$$

$$\left.\frac{\partial^3 h}{\partial x^3}\right|_{n+\frac{1}{2}} = \frac{h_{n+2} - 3h_{n+1} + 3h_n - h_{n-1}}{(\Delta x)^3} - \frac{(\Delta x)^2}{8} \left.\frac{\partial^5 h}{\partial x^5}\right|_{n+\frac{1}{2}} \tag{9.1.92}$$

$$\left.\frac{\partial^4 h}{\partial x^4}\right|_{n} = \frac{h_{n+2} - 4h_{n+1} + 6h_n - 4h_{n-1} + h_{n-2}}{(\Delta x)^4} - \frac{(\Delta x)^2}{6} \left.\frac{\partial^6 y}{\partial x^6}\right|_{n} \tag{9.1.93}$$

In each case the first term on the right is the central-difference approximation and the second term gives the approximate error. Note that the error is proportional to the square of the cell size in each case.

It may be seen from Eqs. (32) to (35) that the same finite-difference approximations are used in the torsion-rod equation with the consequence that here too the errors are proportional to the square of the cell size.

We have presented the finite-difference errors due to approximations in calculating the various derivatives necessary for the solution of beam bending and torsion problems. There remains the question of how these errors are reflected in the calculated mode frequencies.

Michie has obtained analytical solutions for the mode frequencies of a uniform finite-difference beam in bending or torsion for the common types of end conditions.[11] A summary is presented below.

From Eq. (31) we found that for a continuous uniform torsion rod

$$JG \frac{\partial^2 \phi}{\partial x^2} = I \frac{\partial^2 \phi}{\partial t^2} \tag{9.1.94}$$

giving a solution

$$\phi = \sum_k (E_k e^{i\alpha_k x} + F_k e^{-i\alpha_k x}) e^{i\psi_k t} \tag{9.1.95}$$

where

$$\psi_k^2 = \frac{JG}{I} \alpha_k^2 \tag{9.1.96}$$

The amplitude coefficients E_k and F_k are evaluated from initial conditions. The eigenvalues α_k are evaluated from the end conditions.

The solution to the corresponding finite-difference equation is

$$\phi_n = \sum_k (\bar{E}_k e^{ink} + \bar{F}_k e^{-ink}) e^{\pm i\bar{\psi}_k t} \tag{9.1.97}$$

where

$$\bar{\psi}_k^2 = \frac{4JG}{I(\Delta x)^2} \sin^2\left(\frac{k}{2}\right) \tag{9.1.98}$$

Note that the barred quantities are similar to the corresponding unbarred quantities for the continuous case. Also, k is the finite-difference eigenvalue and n is the station number. The correspondences as the cell size becomes very small are

$$\lim_{\substack{n\to\infty \\ \Delta x\to 0}} n\,\Delta x = x$$

$$\lim_{\substack{k\to 0 \\ \Delta x\to 0}} \frac{k}{\Delta x} = \alpha_k$$

The bending deflection for a continuous uniform beam is found from

$$EI\frac{\partial^4 h}{\partial x^4} = -\rho\frac{\partial^2 h}{\partial t^2} \tag{9.1.99}$$

The solution is

$$h = \sum_j (A_j e^{i\lambda_j x} + B_j e^{-i\lambda_j x} + C_j e^{\lambda_j x} + D_j e^{-\lambda_j x})e^{i\phi_j t} \tag{9.1.100}$$

where

$$\phi_j{}^2 = \frac{EI}{\rho}\lambda_j{}^4 \tag{9.1.101}$$

The λ_j are eigenvalues which are evaluated from end conditions.

The corresponding solutions for the bending of a finite-difference beam are

$$h_n = \sum_j (\bar{A}_j e^{inj} + \bar{B}_j e^{-inj} + \bar{C}_j e^{ng(j)} + \bar{D} e^{-ng(j)})e^{i\bar{\phi}_j t} \tag{9.1.102}$$

Table 9.1.3. Beam Bending and Torsion Frequencies

End conditions	Termination	Characteristic equations: Bending	Characteristic equations: Torsion
Cantilever	Actual	$\cos\lambda_j L\cosh\lambda_j L + 1 = 0$	$\alpha_k L = \frac{(2s-1)\pi}{2}$
	Mid-cell	$\cos Nj\cosh Ng + 1 = \frac{\sinh Ng\sin Nj}{\sinh g\sin j}(\cosh g\cos j - 1)$	$Nk = \frac{(2s-1)\pi}{2}$
	End-cell	$\cos Nj\cosh Ng + 1 = \frac{\sinh Ng\sin Nj}{2\sinh g\sin j}(\sinh^2 g - \sin^2 j)$	$Nk = \frac{(2s-1)\pi}{2}$
Built-in or free, both ends	Actual	$\cos\lambda_j L\cosh\lambda_j L - 1 = 0$	$\alpha_k L = s\pi$
	Mid-cell	$\cos Nj\cosh Ng - 1 = \frac{\sinh Ng\sin Nj}{\sinh g\sin j}(\cosh g\cos j - 1)$	$Nk = s\pi$
	End-cell	$\cos Nj\cosh Ng - 1 = \frac{\sinh Ng\sin Nj}{2\sinh g\sin j}(\sinh^2 g - \sin^2 j)$	$Nk = s\pi$
Simply supported, both ends	Actual	$\lambda_j L = s\pi$	$\alpha_k L = s\pi$
	End-cell	$Nj = s\pi$	$Nk = s\pi$
		$s = 1, 2, 3, \ldots$ j = eigenvalue $g = 2\sinh^{-1}\sin\left(\frac{j}{2}\right)$ $\bar{\phi}_j{}^2 = \frac{16EI}{\rho(\Delta x)^4}\sin^4\left(\frac{j}{2}\right)$ where $\bar{\phi}_j$ is mode frequency, radians per sec	$s = 1, 2, 3$ k = eigenvalue $\bar{\gamma}_k{}^2 = \frac{4JG}{I(\Delta x)^2}\sin^2\frac{k}{2}$ where $\bar{\gamma}_k$ is mode frequency, radians per sec

where

$$\bar{\phi}_j{}^2 = \frac{16EI}{\rho(\Delta x)^4}\sin^4\left(\frac{j}{2}\right) \tag{9.1.103}$$

$$g(j) = 2\sinh^{-1}\sin\left(\frac{j}{2}\right) \tag{9.1.104}$$

and j is the finite-difference eigenvalue. Again we note the limits

$$\lim_{j\to 0} g(j) = j$$

$$\lim_{\substack{n\to\infty \\ \Delta x\to 0}} n\,\Delta x = x \qquad \lim_{\substack{j\to 0 \\ \Delta x\to 0}} \frac{j}{\Delta x} = \lambda_j$$

giving

$$\lim_{\substack{j\to 0 \\ n\to\infty \\ \Delta x\to 0}} nj = \lambda_j x \qquad \lim_{\substack{j\to 0 \\ n\to\infty \\ \Delta x\to 0}} ng(j) = \lambda_j x$$

For a specific N-cell beam of total length L, Eq. (97) or (102) can be solved for the eigenvalues, using the end conditions. The results are shown in Table 9.1.3.

Table 9.1.4. Percentage Error in Mode Frequencies

Mode	N	Bending: Cantilever, Mid-cell	Bending: Cantilever, End-cell	Bending: Built-in or free, both ends, Mid-cell	Bending: Built-in or free, both ends, End-cell	Bending: Simply supported, both ends, End-cell	Torsion: Cantilever, Mid-cell or End-cell	Torsion: Built-in or free, both ends, Mid-cell or End-cell
1	2	−13.77	16.72	.....	49.43	18.94	2.55	9.97
	4	−3.36	4.94	−1.14	19.87	5.03	0.64	2.55
	6	−1.48	2.28	−1.06	10.08	2.28	0.31	1.15
	8	−0.83	1.30	−0.70	5.98	1.27	0.18	0.64
	10	−0.51	0.84	−0.47	3.93	0.86	0.10	0.43
	12	−0.37	0.65	−0.34	2.76	0.61	0.07	0.31
2	2	.....	86.71	.....	.....	.....	21.6	36.3*
	4	−1.03	20.62	17.96	36.45	18.94	5.68	9.97
	6	−0.97	10.49	5.61	20.03	8.81	2.55	4.51
	8	−0.64	6.22	2.63	12.34	5.03	1.43	2.55
	10	−0.44	4.09	1.52	8.28	3.26	0.95	1.64
	12	−0.31	2.88	0.99	5.90	2.28	0.59	1.15
3	4	17.94	36.37	.....	52.75	38.50	15.31	21.58
	6	5.59	19.98	17.88	31.02	18.94	6.99	9.47
	8	2.61	12.31	8.98	19.80	11.00	4.16	5.68
	10	1.51	8.26	5.37	13.55	7.18	2.55	3.66
	12	0.99	5.89	3.57	9.78	5.03	1.79	2.55
4	4	.....	52.76	.....	.....	.....	28.64	36.34*
	6	17.89	31.03	34.10	42.69	32.61	13.42	17.30
	8	8.99	19.80	18.06	28.16	18.94	7.67	9.47
	10	5.37	13.55	11.02	19.65	12.49	4.97	6.45
	12	3.57	9.79	7.39	14.37	8.81	3.46	4.51

$$\%\ \text{error} = 100 \times \frac{\text{actual frequency} - \text{finite-difference frequency}}{\text{actual frequency}}$$

* Mid-cell termination, free, and end-cell termination, built-in, omit this mode.

A comparison can be made of the mode frequencies for the continuous and finite-difference beams. The percentage error in mode frequencies is given in Table 9.1.4 for various end conditions and cell numbers. Some of these results are plotted in Fig. 5.10.18.[3]

9.1.21. Parasitic Impedances. One of the most troublesome sources of error in the direct-analogy computer lies in the parasitic impedances of the computing and metering elements. By *parasitic impedances* we mean the imperfections of the actual elements compared with the ideal, aside from calibration errors or nonlinearities. Such things as series resistance or shunt capacitance in inductors or transformers are examples. The effect of the parasitic impedances is to produce additional degrees of freedom and also to add damping to the circuit.

For the usual sorts of structural and dynamics problems of moderate size such as we have been considering, the errors due to parasitic impedances can be made relatively small. It is quite important in the reduction of these errors that the circuit be properly scaled. Also, compensation can often be achieved by slight adjustments in the settings of the circuit elements.

The principal offenders as far as parasitic impedances are concerned are the transformers and inductors. These elements are especially constructed for use in computers in order to be as nearly perfect as possible.

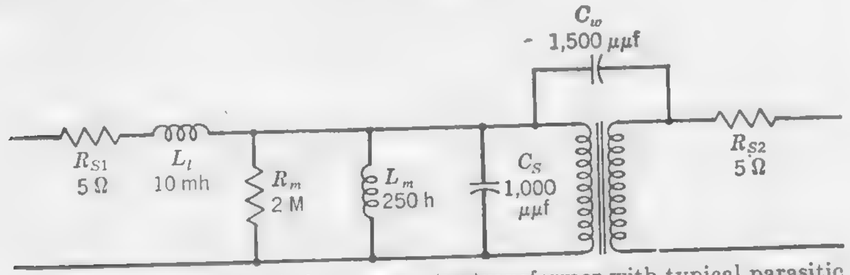


FIG. 9.1.33. An equivalent circuit for a computer transformer with typical parasitic impedance values.

A somewhat simplified equivalent circuit of a computer transformer is shown in Fig. 9.1.33 with typical values for the parasitic elements. Each transformer winding has many taps in order that a fine adjustment can be made on the turns ratio. As the turns ratio is varied the parasitic impedances except for R_{s1} and R_{s2} will follow the usual transformation law; i.e., the impedance varies as the square of the turns ratio. The winding resistances R_{s1} and R_{s2} will vary directly with the turns in the corresponding winding. For the usual computer frequencies and impedance levels, neither the series nor shunt impedances will cause appreciable error provided that the turns ratio is near unity. But because of the square-law variation of impedances with turns ratio, large errors can be introduced by wide usage of turns ratios of the order of 0.1.

Even though the effects of parasitic impedances may be relatively small in a given case, it is often worthwhile to compensate for certain parasitics by adjusting element settings. For example, a transformer shunt capacitance can be compensated by setting a shunt capacitance in the network to a correspondingly low value. Series impedances such as leakage inductance can be compensated by altering the setting of a corresponding series element.

Many of the transformer parasitics have mechanical analogies which are often present in an actual mechanical system. In these cases the effect of the parasitic impedance is to alter the mechanical system in a physically understandable manner. For example, in the case of a simple beam circuit the following correspondences occur: (1) Leakage inductance L_l represents shear compliance. (2) Magnetizing inductance L_m represents tension in the beam. (3) Shunt capacity C_s represents rotary inertia. (4) Series resistance R_s represents damping in shear.

The inductor parasitic impedances of major interest are the effective series resistance and the shunt capacitance. The series resistance is a function of frequency and is

most often described in terms of the parameter

$$Q = \frac{\omega L}{R} \tag{9.1.105}$$

where R is the effective series resistance. The Q of a computer inductor varies with inductor size, tap setting, frequency, and current. For a typical 1-henry inductor the peak value of Q will be near 500 and it will exceed 100 over the normal range of frequencies (100 to 2,000 cps) for all tap settings.

Shunt parasitic capacitance in an inductor will cause the effective inductance to increase for all frequencies below the self-resonant frequency. The effective inductance is

$$L_{eff} = \frac{L}{1 - (\omega/\omega_0)^2} \tag{9.1.106}$$

where L is the low-frequency inductance and ω_0 is the self-resonant frequency (radians per second). The self-resonant frequency of a typical 1-henry inductor is about 10,000 cps, but larger inductors may resonate as low as 3,500 cps.

The parasitic impedances in the resistors and capacitors are negligible for the usual high-quality components. The resistors are usually of the wire-wound type while the capacitors have polystyrene or mica as the dielectric.

Other parasitic impedances which need to be considered are the shunt wiring capacitances and metering impedances. Stray wiring capacitance of the order of several hundred micromicrofarads might arise, for example, when one is interconnecting elements in separated portions of the computer. The size of the shunt metering capacitance will depend upon size and design details of the specific computer but would typically be greater than 0.001 μf. This capacitance can influence the results of voltage measurements and is connected between the two points being metered.

Another metering impedance that can influence accuracy is the series resistance inserted in the circuit at the time of a current measurement. One can normally avoid these errors by choosing the proper size of metering shunt.

References

1. Benscoter, S. U., and R. H. MacNeal: Introduction to Electrical-circuit Analogies for Beam Analysis, *NACA Tech Note* 2785, 1952.
2. Benscoter, S. U., and R. H. MacNeal: Equivalent Plate Theory for a Straight Multicell Wing, *NACA Tech. Note* 2786, 1952.
3. Howe, C. E., and R. M. Howe: Application of Difference Techniques to the Lateral Vibration of Beams Using the Electronic Differential Analyzer, *Univ. Mich. Dept. Aeronaut. Eng. Rept.* AIR-7, 1954.
4. MacNeal, R. H.: The Equivalent Circuits of Shells Used in Airframe Construction, *Proc. Western Computer Conf.*, 1953.
5. MacNeal, R. H.: Vibrations of Composite Systems. *Calif. Inst. Technol., ARDC Rept.* 4, 1954.
6. MacNeal, R. H.: The Solution of Elastic Plate Problems by Electrical Analogies, *J. Appl. Mechanics*, **18**(11):59–67 (1951).
7. MacNeal, R. H., and G. D. McCann: Ideal Transformers in the Synthesis of Analog Computer Circuits, *Proc. Western Joint Computer Conf.*, pp. 16–23, 1955.
8. MacNeal, R. H., G. D. McCann, and C. H. Wilts: The Solution of Aeroelastic Problems by Electrical Analogies, *J. Aeronaut. Sci.*, **18**:777–789 (1951).
9. McCann, G. D., and R. H. MacNeal: General Methods of Synthesis for Electrical-mechanical Analogies of Aircraft Structures, *Calif. Inst. Technol., ARDC Rept.* 1, 1954.
10. McCann, G. D., and R. H. MacNeal: Beam Vibration Analysis with the Electric Analog Computer, *J. Appl. Mechanics*, **17**:13–26 (1950).
11. Michie, R. W.: Frequency Errors in Analog Computer Methods, *Lockheed Aircraft Corp. Rept.* MAS-46, 1956.
12. Russell, W. T.: Lumped Parameter Analogies for Continuous Mechanical Systems, California Institute of Technology *Ph.D. Thesis*, 1950.
13. Russell, W. T., and R. H. MacNeal: An Improved Electrical Analogy for the Analysis of Beams in Bending, *J. Appl. Mechanics*, **20**:349–355 (1953).

PART 2: DIRECT-ANALOGY COMPUTERS: COMPONENTS AND SYSTEMS

By William J. Dixon

INTRODUCTION

9.2.1. General Requirements of a Direct-analogy Computer. A network-type direct-analog computer for the solution of partial differential equations should incorporate the following components and features:

1. *Passive elements* (transformers, inductors, resistors, capacitors) to simulate the energy-conserving and energy-dissipating parts of the physical system being analyzed. These elements must be capable of being set easily and accurately.
2. *Active elements* (amplifiers, multipliers, function generators) to simulate the energy-supplying parts of the physical system. In addition, nonlinear effects and time-varying parameters are achieved through the use of active elements.
3. Means for convenient *interconnection* of the passive and active elements into circuits of any desired configuration. This is generally accomplished by means of removable patchboards.
4. A method of *exciting* the analog network by applied voltages and/or currents must be provided in the computer. In general the excitation of the network either is transient in nature or consists of steady-state sinusoidal functions. (In analogies employing all-resistor networks, such as the solution of Laplace's equation, the steady-state excitation may be d-c instead of a-c.)

Table 9.2.1. Examples of Direct-analogy Computer Problems Employing Passive-element Networks and Active Elements

Type of problem	Simulation achieved by passive-element network	Simulation achieved by active elements
Dynamic aeroelastic system (flutter)	Structural properties: Elastic forces Inertia forces Damping forces	Aerodynamic forces
Missile autopilot system	Missile structural properties Mechanical characteristics of gimbaled engines, control surfaces, etc. Fuel sloshing	Autopilot computer Servo-motor characteristics Aerodynamic forces
Heat transfer	Heat capacity Conductive heat transfer	Radiation heat transfer Heat transfer to a moving fluid Change of phase (freezing, boiling) Energy supplied by combustion, nuclear reaction, etc.
Nuclear-reactor kinetics	Neutron flux level and absorption Delayed neutron flux level and absorption Reactor-temperature build-up	Production of neutrons Generation of heat Dependence of reactivity on temperature

Generally, the systems simulated by purely passive-element networks are those in which energy is conserved or dissipated but in which there are no internal energy sources. (In the above examples, energy is introduced as an initial condition, or during the solution time from an external source.)

In many instances, it is desirable to simulate a system which is composed of both passive and active subsystems. For example, in an aircraft flutter problem, the dynamic structural characteristics of the airframe are readily amenable to simulation by a passive-element network, but the proper representation of the aerodynamic forces requires energy-supplying components. (The existence of instability in an aeroelastic system is sufficient evidence for this requirement.) Table 9.2.1 shows examples of problems in which the analog circuit consists of both passive-element networks and active elements.

PASSIVE ELEMENTS

9.2.2. Selection of Frequency Range, Impedance Level, and Power Level. The range of working frequencies and useful network-impedance values in any direct-ana-

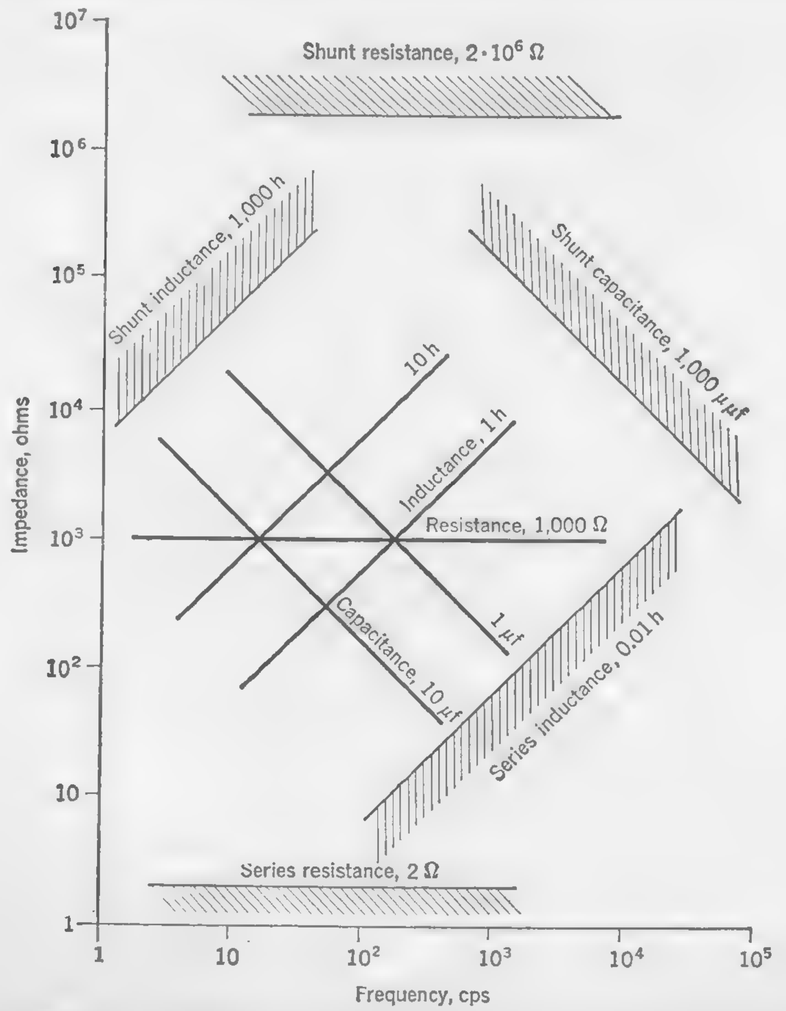


FIG. 9.2.1. Limitations imposed on impedance and frequency ranges by parasitic effects. The locations of the shaded lines indicate where specific classes of parasites would affect performance of the analog network. Values chosen for the amount of these parasites (in parentheses) are not associated with any particular computer but have typical order of magnitude. See Table 9.2.2.

log computer is limited by *parasitic impedances* associated with network elements, active computing elements, peripheral equipment, and interconnections.

One can often compensate for parasitic effects in passive networks by adjusting the values of inductance, capacitance, or resistance to which elements are set. If, for example, the analog circuit requires an inductor to be connected in parallel with a transformer winding, the calculated value of the inductor may be increased to compensate for the transformer's shunt magnetizing inductance.

Figure 9.2.1 shows how various classes of parasitic effects tend to limit the operation of passive-element computers to certain ranges of impedances and frequencies. In that figure, in which the axes logarithmically represent impedance and frequency, the approximate limitations of the useful range are indicated by shaded lines around the periphery. The solid lines in the interior of the diagram correspond to typical values of computer elements. A summary of the sources of the parasitic effects which are illustrated in Fig. 9.2.1, and typical values found in the components and installations of direct-analogy computers, are given in Table 9.2.2.

In general, the usefulness and versatility of a passive-element computer increase with the ratio of maximum impedance to minimum impedance which it can accommodate, and also increase with the ratio of maximum frequency to minimum frequency. Consequently, a general design goal is to occupy as much area on a log impedance–log frequency plot, such as Fig. 9.2.1. Of course, some computers of a special-purpose nature are designed to operate at only a single frequency, for example:

1. Transformer-resistor networks for the solution of static elastic structures (Sec. 9.1)
2. Transformer-inductor-resistor-capacitor networks to simulate steady-state characteristics of power-distribution systems (Sec. 9.3)
3. All-resistor networks, operating at a fixed frequency, or at direct current

In the first two examples given, it is feasible to compensate for the existence of the reactive (inductive or capacitive) parasitic effects. For instance, wherever a shunt inductive parasite exists, a shunt capacitor would be installed and tuned to give parallel resonance at the operating frequency. The useful impedance range of a single-frequency network can thus be extended to limits imposed by purely resistive parasitic effects.

From the data presented above, it will be noted that no passive-element network making use of magnetic components (transformers and inductors) will operate at zero frequency. The transformer has a shunt inductive parasite, and no matter how large this inductance is kept, at some low frequency ideal transformer action will be compromised by it. In the case of the inductor, at very low frequencies its series resistance will be comparable with the desired inductive reactance. At very low frequencies, even the leakage resistance of the capacitors would be comparable with their capacitive reactance. However, the state of capacitor design is such that leakage time constants of the order of days, or even months, can be obtained. Therefore, for all practical purposes, resistor-capacitor networks can be used down to zero frequency.

In establishing the operating range of the direct-analogy computer, one must also specify the *power level* of signals in the passive-element networks. Note the following limitations on power levels:

Maximum permitted voltage for capacitors
Rated power dissipation for resistors
Maximum flux density for linear operation of inductors and transformers

For transformers, the limitation would be stated in terms of the ratio of full-winding voltage to frequency. For inductors, the limitation is best stated as a maximum current. These limitations are more for the purpose of avoiding operation in a nonlinear range of the magnetic materials than to protect the elements.

In addition, intentional nonlinearities are frequently introduced in the direct-analogy computer. It is necessary that the power level existing in the network be related properly to the level at which the nonlinear devices exhibit their characteristic behavior. However, the most common use of nonlinear devices in direct-analogy computers is in conjunction with the active elements, and it is generally adequate that the passive elements and active elements have compatible power-level ranges.

Table 9.2.2. Parasitic Effects Influencing Impedance and Frequency Ranges of Direct-analogy Computers
The values given are typical of the performance of components in computer installations

Parasitic effects associated with passive elements		Transformer	Inductor (1 henry)	Resistor	Capacitor (1 μf)	Other parasitic effects
Shunt parasitic effects[a]	Inductive parasite	Magnetizing inductance, 200 henrys				Magnetizing inductance of metering isolation transformer, 10,000 henrys
	Resistive parasite	Hysteresis and eddy-current loss, 2 megohms	Hysteresis and eddy-current loss, 2 megohms		D-C leakage resistance, 100,000 megohms Dielectric loss at 500 cps, 1 megohm	Insulating resistance, wiring and plugboards, 1,000 megohms Metering system input impedance, 1 megohm Current-generator output impedance, 1 megohm
	Capacitive parasite	Capacitance of both coils, 1,700 $\mu\mu$f	Capacitance of coil, 300 $\mu\mu$f	Negligible		Passive-element wiring capacitance, 100 $\mu\mu$f Bus-to-bus and bus-to-ground capacitance, 500 $\mu\mu$f Metering-system input impedance, 1,000 $\mu\mu$f Current-generator output impedance, 1,000 $\mu\mu$f
Series parasitic effects	Inductive parasite	Leakage inductance, 10 millihenrys		Negligible	Negligible	Lead and bus self- and mutual inductance, 20 microhenrys
	Resistive parasite	D-C resistance of coils, 11 ohms	D-C resistance of coil, 2 ohms		Negligible	Decade switch contacts, relay contacts, element leads, 0.1 ohm Bus series resistance, 0.1 ohm Shunt used in metering currents, 1 ohm
	Capacitive parasite					

For active elements (amplifiers, current generators, photoformers, and multipliers), the limit of operating level is usually stated in terms of both maximum voltage and maximum current which can be supplied by the output stage.

The power level existing in the passive-element network must be large enough so that the metering-system impedances do not introduce significant errors. Finally, the existence of thermal noise in electric circuits imposes a lower limit on the operating

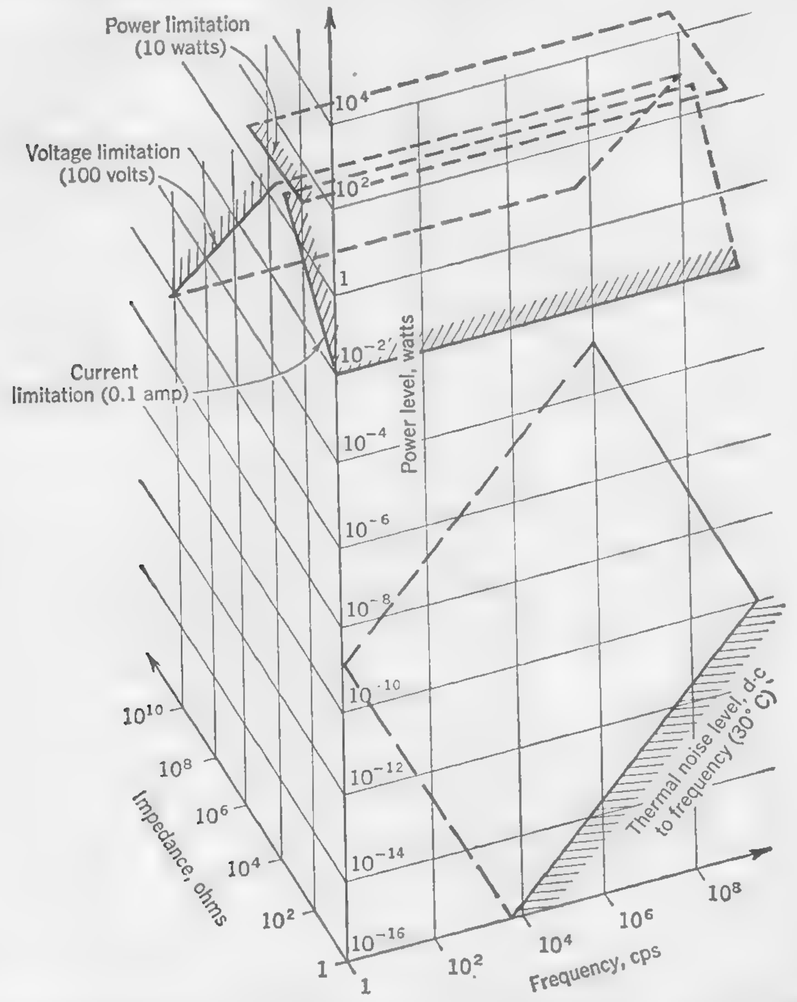


FIG. 9.2.2. Limitations imposed on power level. The upper voltage, current, and power limitations are typically the correct order of magnitude of the capabilities of computer components. The thermal noise level is the theoretical value for a pure resistance; use of vacuum-tube devices with high input impedances will give noise in excess of this amount.

level of the direct-analogy computer. The greater the frequency band over which the computer operates, the greater will be the amount of thermal noise which exists.

Figure 9.2.2 summarizes the nature of limitations on power level and indicates the upper level imposed by typical limitations on voltage and current in both passive and active components and a lower limitation given by the theoretical level of thermal noise.

It should be noted that the effects of a parasite are not necessarily based on a comparison of the magnitude of the parasitic effects and the values to which the passive elements are set. For example, the evaluation of the effect of a shunt parasite existing between a particular node of the network and ground should be based on a comparison of the magnitude of the parasite and the input impedance of the passive-element network at that node. The input impedance may vary considerably from node to node and may bear little relation to the values to which passive elements are set in

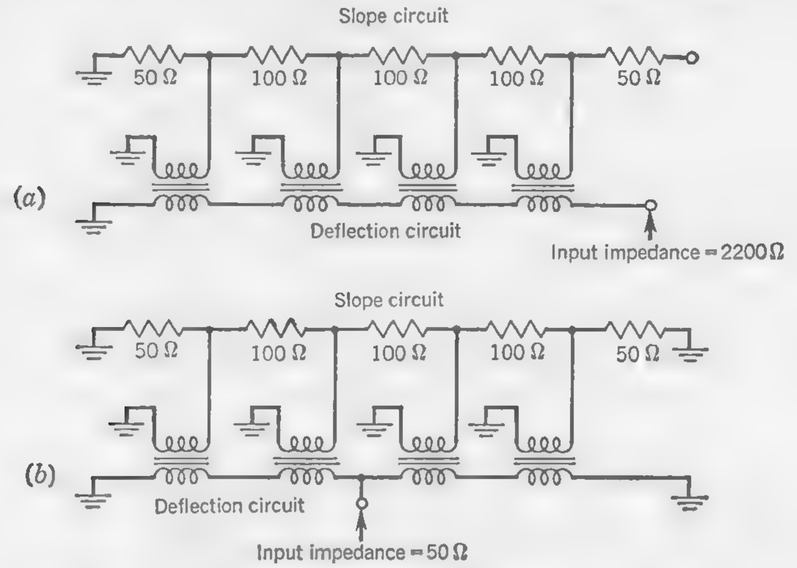


FIG. 9.2.3. Illustration of the difference between the impedance of the passive elements and the impedance level of the circuit. Circuits (*a*) and (*b*) each represent the static analog of a four-cell uniform elastic beam. Each transformer has a 1:1 turns ratio. In (*a*) the beam is clamped at one end, and the input impedance (in the deflection circuit) at the free end is 2,200 ohms. In (*b*) the same beam is clamped at both ends, and the input impedance (in the deflection circuit) at the center of the beam is 50 ohms.

the circuit. As an illustration of this, Fig. 9.2.3 shows two circuits composed of identical circuit elements, with resistor values of 50 and 100 ohms. Yet the input impedance in one example is 2,200 ohms and in the other it is 50 ohms. Thus an impedance limitation imposed by the existence of parasites may apply to the impedance level of the entire circuit as characterized by values of input impedance at the nodes of the circuit, as well as to the values of the individual passive elements.

In a typical direct-analogy computer employed in static and dynamic structural analysis, the passive elements will have the following nominal maximum values:

Inductors: 80 to 90 per cent of the elements, 1.0 henry. Remainder, 10 henrys

Resistors: 80 to 90 per cent of the elements, 10,000 ohms. Remainder, 100,000 ohms

Capacitors: 80 to 90 per cent of the elements, 1.0 μf. Remainder, 10 μf

9.2.3. Transformers and Inductors. The importance of the *transformer* as an element in direct-analogy computers is evident in the circuitry developed in Sec. 9.1. The extension of direct-analogy techniques into the solution of wide varieties of complex structural systems has been essentially based on the continued development of *high-quality computing transformers*.

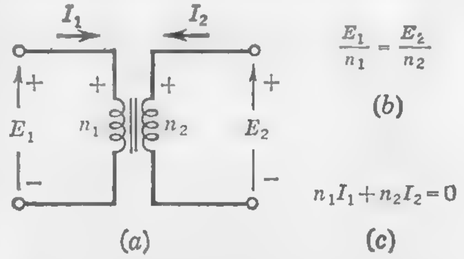


FIG. 9.2.4. Ideal transformer. (*a*) is the symbol of an ideal two-winding transformer. n_1/n_2 is the ratio of the number of turns in coil 1 to the number of turns in coil 2. (*b*) and (*c*) are the equations governing the behavior of the ideal transformer. To achieve equation (*b*) would require a complete linkage of all the magnetic flux by each turn of each winding. (*c*) would require an infinitely permeable magnetic core material. Polarity is given by the + signs in (*a*), indicating geometrically similar ends of the two windings.

The goal in transformer design, in the case of a two-winding transformer, is indicated by the "ideal transformer" in Fig. 9.2.4. In a transformer with more than two windings—say m windings—the ideal equations are generalized as follows:

$$\frac{E_1}{n_1} = \frac{E_2}{n_2} = \frac{E_3}{n_3} = \cdots = \frac{E_m}{n_m}$$

$$\sum_{i=1}^{m} n_i I_i = 0$$

To obtain transformer behavior approaching these equations one must

1. *Minimize* leakage flux, stray capacitance between coils, wire resistance, and core losses
2. *Maximize* the permeability of the core material

The best computer transformers available are based on a spirally wound core of supermalloy. The windings are applied in toroidal layers. The transformer is hermetically sealed to preserve the dielectric properties of the insulation. The resulting transformer package is about 9 in. in diameter and 5 in. high.

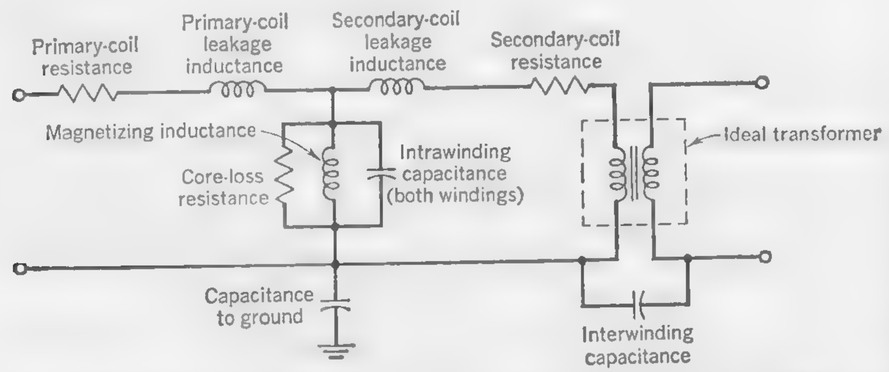


FIG. 9.2.5. Approximate equivalent circuit of actual two-winding transformer. All the parasitic effects have been referred to the primary winding. If the parasitic effects are negligibly small, the circuitry reduces to that of the ideal transformer.

Figure 9.2.5 gives the approximate equivalent circuit of an actual transformer. The magnitudes of the parasitic elements indicate the extent of deviation from the ideal. The first column of Table 9.2.2 gives typical parasite magnitudes of a top-quality transformer, with a turns ratio of 1:1, and all windings connected in series.

When installed in a computer, provision is made for connecting the windings in series or in parallel (to minimize shunt or series parasitic effects, respectively) and for selecting the proper portion of the total number of turns available. This may be accomplished by means of switches and plugboards. The form of such connections is illustrated in the example of Fig. 9.2.6.

In the design of *inductors* it is desired to keep the Q large and to keep shunt capacitance low. Assuming that the inductor carries a current which varies sinusoidally with time, Q, the storage factor, is given by the equation

$$Q = 2\pi \frac{\text{peak magnetic energy stored by inductor}}{\text{energy dissipated in one cycle}}$$

If the only dissipating effect is series resistance R, then:

$$Q = \frac{\omega L}{R} = \frac{2\pi f L}{R}$$

At low frequencies the Q is influenced most by the d-c resistance of the winding, a series parasitic resistance. At high frequencies the Q is determined principally by losses in the magnetic core. Core-loss effects are best characterized by attributing them to an equivalent shunt resistance. Thus the maximum Q will exist at some intermediate frequency, at which the inductive reactance ωL is equal to the geometric mean of the series resistance and shunt resistance of the inductor.

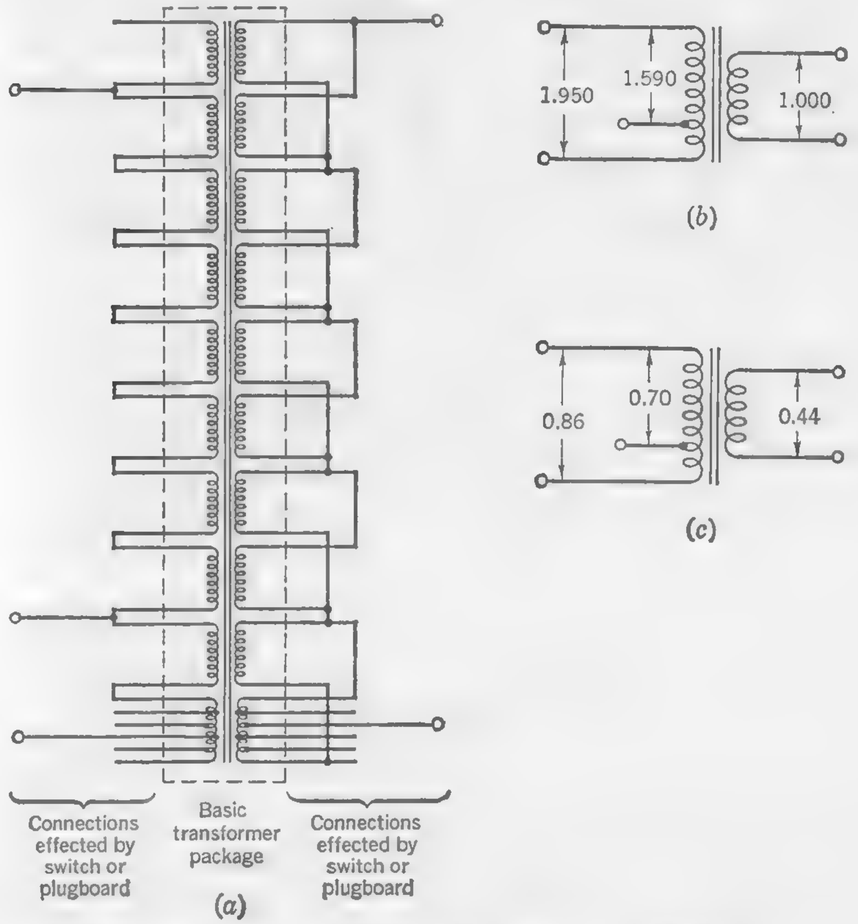


FIG. 9.2.6. Transformer connections. In the basic transformer package both the primary and secondary have 10 winding segments, each with one-tenth the total number of turns. The tenth segment has leads available at four intermediate (2 per cent) points. In (*a*) the primary winding is connected in series to minimize shunt parasitic effects, and the secondary winding is connected in parallel to minimize series parasitic effects. (*b*) shows an example of desired transformer turns ratio and tap location. (*c*) shows how the connections of (*a*) are chosen (in fractions of the entire windings) to approximate the desired values.

The best passive-element computer inductors make use of ferrite cores. Others make use of permalloy dust cores. The most convenient arrangement is to obtain all inductance values for one decade by switching to different taps of the coil of a single inductor. The inductance for a given core is proportional to the square of the number of turns. Consequently, in order to switch inductances of 0.1, 0.2, 0.3, . . . , 0.9, 1.0 times the full inductance of the decade, it is necessary to switch 0.316, 0.447, 0.548, . . . , 0.949, 1.000 times the full number of turns. A computer inductor will be made up of three such decade inductors connected in series, with the top decade having a full value of 1.0 henry or 10 henrys.

A plot of Q vs. frequency for a 1.0-henry decade inductor is shown in Fig. 9.2.7. In terms of series and shunt parasites, this inductor corresponds to the data given in Table 9.2.2.

9.2.4. Resistors. The design of computer resistors is the least challenging of the elements, parasitic effects being, for the most part, negligible.

The principal cautions to be observed are:

1. Avoiding excessive shunt capacitance
2. Minimizing the variation of resistance with changes in temperature
3. Minimizing the variation of resistance with changes in applied voltage
4. Achieving stability of resistance over long periods of time

The use of wire-wound resistors employing wire material with low temperature coefficient of resistivity is satisfactory in all these respects. A 10-ohm resistor decade is constructed by connecting ten 1-ohm wire-wound resistors in series. The decade switch spans as many of the series resistors as its position indicates. Similar construction is employed for resistor decades of 100 ohms, 1,000 ohms, etc. A computer resistor generally consists of three or four decade resistors. The top decade may have a maximum value of 10,000, 100,000, or 1,000,000 ohms.

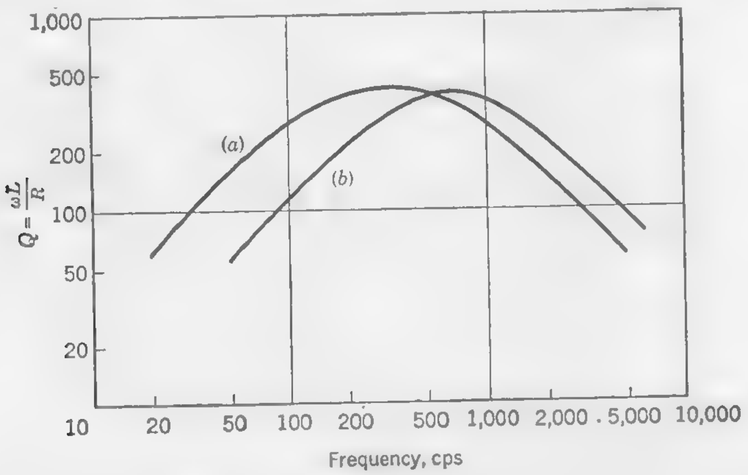


FIG. 9.2.7. Inductor Q vs. frequency. Both curves are for a ferrite-core 1-henry inductor. (*a*) is for the 1.0-henry tap. (*b*) is for the 0.1-henry tap.

In some low-frequency high-impedance computers constructed principally for use in simulating thermal systems, computer resistors as high as 100 megohms have been employed. At such high levels of resistance, it is necessary to be careful to avoid shunt parasitic resistance in such places as decade-switch insulation, bus and element lead insulation, and plugboard insulation. Shunt resistances of 10^9 ohms in such locations would not be noticed in normal computer applications but would limit the accuracy of the high-impedance computer.

9.2.5. Capacitors. It is possible to achieve a fairly close approximation to ideal-capacitor behavior over a wide range of frequencies. The important points in selection of capacitors are:

1. Low dissipation factor (high shunt resistance)
2. Low d-c leakage (high shunt resistance)
3. Low variation in capacitance with frequency

The behavior of the capacitor with regard to these criteria is principally a function of the dielectric material. Polystyrene is superior to other common dielectric materials in these respects and is generally used in the important decades (1.0 and 0.1 μf maximum values). Mica capacitors are used in 0.01-μf decades, and Mylar capacitors are used in 10-μf decades. Mica gives rise to a variation in capacitance between d-c and a-c frequencies, because of a slow recovery from polarization of crystalline interfaces, but restricting its use to the lowest decade renders this effect insignificant.

Typical shunt-resistance parasites of a 1.0-μf polystyrene capacitor are given in Table 9.2.2.

The usual construction of a decade capacitor with maximum value of, say, 1.0 μf makes use of four individual capacitors, with values of 0.1, 0.2, 0.3, and 0.4 μf. By appropriate wiring of a two-deck switch, the correct capacitors are selected and connected in parallel to give all values from 0 to 1.0 μf in 0.1-μf steps. A three-decade capacitor unit is composed of three single-decade capacitors connected in parallel.

ACTIVE ELEMENTS AND NONLINEAR ELEMENTS

9.2.6. Active Elements in General. Mathematically, the use of active elements in the direct-analogy computer is very similar to their use in the electronic analog computer. The principal exception arises from the fact that in the passive-element networks of the direct-analogy computer electric current, as well as voltage, is significant and must be sensed without perturbation; again, outputs produced by active elements may be either currents or voltages. Hence the active elements may have to work into a much wider range of load impedances than those encountered in electronic analog computers.

Whereas electronic analog computers commonly utilize most of their full ±100-volt range, a passive-element network is frequently operated with voltages no greater than 5 volts at a large number of the nodes of the network. This is partly due to the fact that scale changes are commonly introduced at only a few discrete points in the passive-element network. In a typical analogy, a large range of voltages exists between the different parts of the network. In addition, operation within the linear range of transformers and inductors imposes a limit on the voltages which may be impressed across these elements, particularly at low frequencies. As a consequence of the operation of active elements at lower voltage levels, the permissible noise and drift levels generated by these devices are somewhat lower than in the case of the electronic analog computer.

A second factor in the design of the active elements arises from their use over a considerably wider frequency band than is employed by many electronic analog computers. The requirement of maintaining satisfactory behavior out to frequencies of 1,000 cps or greater calls for a design maintaining a high open-loop gain out to higher a-c frequencies than in the usual operational-amplifier design. The greater bandwidth of operation of the amplifier also makes it more difficult to achieve the desired low noise figures.

9.2.7. Amplifiers. The use of amplifiers as computing elements in direct-analogy computers is restricted to the representation of effects which cannot be accounted for by passive elements. Energy sources, some nonlinear characteristics, unilateral transfer impedances or admittances, and several other effects (generally known as "unrealizable" in passive-network theory) cannot be simulated by passive elements.

The most useful general application of amplifiers would be the simulation of transfer impedances or transfer admittances. In a loop analogy, a general *transfer-impedance* circuit inserts in one loop an increment of voltage which depends on the current flowing in another loop. In a nodal analogy, a general *transfer-admittance* circuit inserts into one node a current which depends on the voltage which exists at another node. Figures 9.2.8 and 9.2.9 show how these transfer circuits are constructed. By means of these circuits, or circuits derived from them, unilateral transfer impedances and admittances may be represented, negative self-impedances may be devised, and practically any linear impedance or admittance function, realizable or "unrealizable," may be simulated.

Figure 9.2.8 shows a circuit for a general *transfer impedance* in a loop analogy. The current flowing in the ith loop flows through an impedance Z, producing a voltage I_iZ. In order to avoid the insertion of a voltage in the ith loop (which would disturb the circuit in which the current I_i is being sensed) it is necessary to subtract out the voltage I_iZ. This is done by the use of operational amplifiers with a net gain of $+1$; the transformer in the circuit isolates the two loops from each other. One of the disadvantages of the circuit of Fig. 9.2.8 is the fact that in each of the amplifiers the common

terminal must be "floated" above ground potential, so that separate power supplies may be required. This need for floating amplifiers provides strong motivation for using node rather than loop analogies.

Figure 9.2.9 illustrates the circuit for a general *transfer admittance* in a nodal analogy. In order to avoid disturbing the circuit at the ith node, the input impedance to the amplifier must be large compared with network impedance levels. Frequently, the functions of the amplifier, which isolates the active-element portion of the circuit from the passive-element network, and the transfer function $F(p)$ are combined in the use of a high-gain operational amplifier. Many of the techniques of electronic analog computers which are described in earlier sections of this handbook may be utilized in circuits based on Fig. 9.2.9. The use of inverting amplifiers, integrators, and function

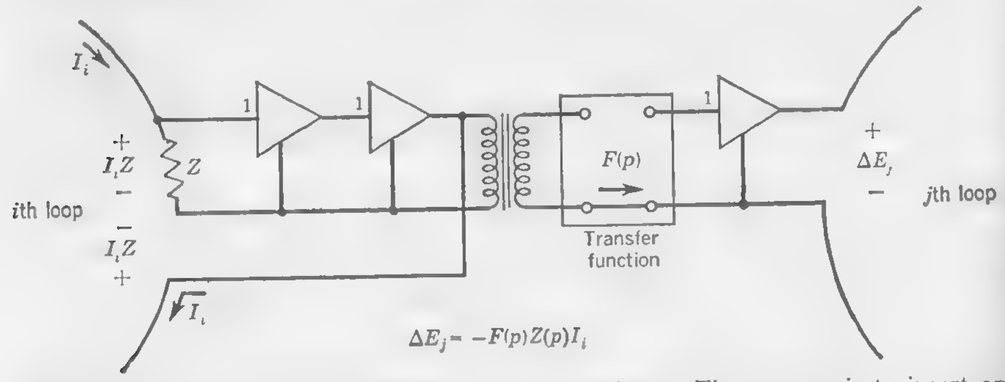


FIG. 9.2.8. General transfer impedance in a loop analogy. The purpose is to insert an increment of voltage ΔE_j in one loop which depends on the current I_i which exists in another loop. $Z(p)$ and $F(p)$ may be considered functions of the operator $p = d/dt$, or of s, the Laplace-transform variable. Note that the common terminal of each amplifier is "floating" above ground.

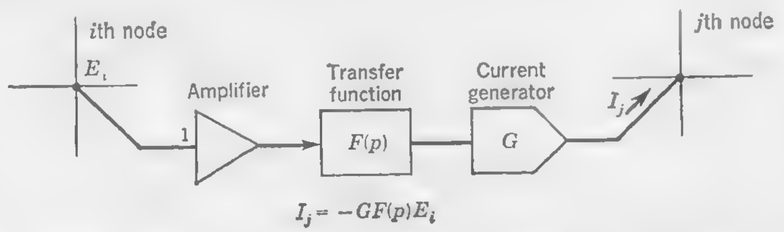


FIG. 9.2.9. General transfer admittance in a nodal analogy. The purpose is to insert a current I_j into one node which depends on the voltage E_i which exists at another node.

generators to operate on the voltage before applying it to the current generator G is obvious. In some instances, it is desired that the current I_j to be inserted at the jth node be the sum of terms, each of which depends upon the voltage at a different node. In this instance, the use of a computer amplifier as a summer is customary.

Figure 9.2.10 shows several *current-generator circuits* using computer amplifiers, together with the output current in terms of the input voltage and the equivalent output impedance of each circuit. Desired values of the transconductance of current generators are typically of the order of 0.001 mho. Assuming the load impedance Z_L is sufficiently small, the transconductance of the circuit of Fig. 9.2.10*a* is $-A/R$. For Fig. 9.2.10*b* the transconductance is $-1/R$, and for (*c*), it is $\pm 1/R$. The ideal current generator delivers a current independent of the load impedance Z_L. The degree to which an actual current generator deviates from this ideal behavior is attributed to the output impedance as illustrated in Fig. 9.2.11. For use in direct-analogy computers it is desirable that output impedances of 1 megohm be attained by the current generators. In most current-generator circuits the output impedance depends on the gain of the amplifiers. Because the amplifier gain is not constant with changes in fre-

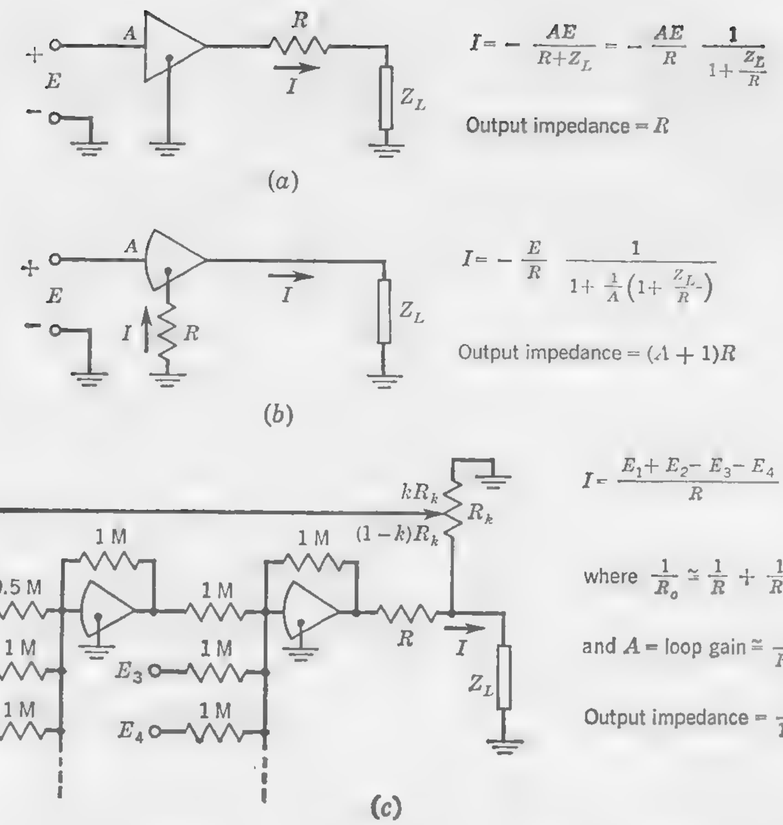


FIG. 9.2.10. Current generators. In each circuit it is desired to generate current I, to be inserted into the node at the top of Z_L. Z_L, the load impedance, represents the input impedance of the circuit receiving the current. (*a*) Current generator composed of voltage amplifier and series resistor. For proper operation R must be large compared with Z_L. To achieve adequate gain, A must be large, and reasonably large voltages appear at amplifier output. (*b*) Voltage amplifier with current feedback through R. Operation and output impedance are satisfactory. Amplifier must "float." (*c*) Summer and current generator composed of voltage amplifiers with voltage feedback. Potentiometer k is set to give unity loop gain to achieve proper operation with maximum output impedance. This circuit will sum voltages in separate channels with positive and/or negative gain, and generate current. The equations for A and R_o are approximate only because of the loading of the potentiometer by the 0.5-megohm resistor, and because of the noninfinite gain of the amplifiers. Typical values of R and R_k are 1,000 and 20,000 ohms, respectively.

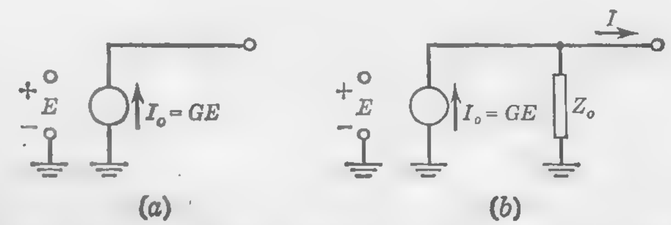


FIG. 9.2.11. Current generator output impedance. (*a*) represents the performance of an ideal current generator. (*b*) indicates how, in an actual current generator, the dependence of output current I on the load impedance may be attributed to the output impedance Z_o.

quency, the output impedance is also a function of frequency. For most current-generator circuits the variation of output impedance with frequency is characterized by the existence of a capacitive component as well as a resistive component.

The current generators of Fig. 9.2.10*b* and *c* are the ones most frequently used in direct-analogy computers; (*b*) has the advantage of simplicity but requires a floating amplifier with a separate power supply; (*c*) has the advantage that it fulfills the functions of summation and current generation. Another useful feature of this current-generator circuit is the fact that the output impedance may be controlled by the potentiometer setting the loop gain. It is, indeed, possible to obtain a *negative impedance*, say to compensate for undesired shunt parasitic effects existing in the passive-element network. A controlled negative output impedance achieved in this manner will be predominately resistive.

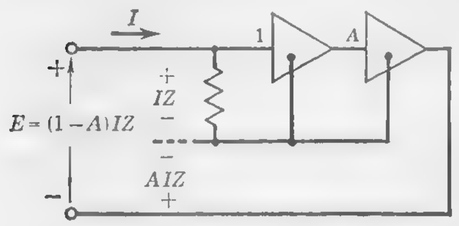


FIG. 9.2.12. Two-terminal negative impedance. The impedance of this circuit is $(1 - A)Z$ and is negative when A is greater than unity. Z may be any two-terminal passive-element network. Note that the amplifiers must float.

By means of a similar circuit, shown in Fig. 9.2.12, a general *negative two-terminal impedance* may be simulated. The need for such simulation does not arise very frequently, and the value of this circuit is more mathematical than practical.

The active-element networks presented above either have current inputs and voltage outputs or voltage inputs and current outputs. Active-element circuits with voltage inputs and voltage outputs would be based on electronic-computer usage. However, direct-analogy computers do not make frequent use of this sort of simulation. The reason is that most equations giving one voltage as a function of other voltages are in one of these two classes:

1. The stated relation involving several voltages expresses a physical constraint. In such cases there is always a similar relation involving the currents of the passive-element network. This relation is either stated or implicit. To the extent that this constraint is linear it can always be represented by a system of computer transformers, in which case the implied current relations will automatically be satisfied.
2. The equation relating several voltages does not represent a physical constraint, but it does represent the solution of network equations involving impedances. In this case there are usually intermediate equations involving network currents which have been omitted. If these intermediate equations are reconstructed, the proper simulation is generally achieved by the use of passive elements, or by the use of active-element circuits representing transfer admittances or impedances.

As noted at the beginning of this article the design requirements of active elements for the direct-analogy computer are somewhat stricter than electronic-analog-computer requirements in the following categories:

1. The bandwidth requirements are greater,
2. The drift and noise level should be lower.

The connection of computer amplifiers to perform the function of current generation also calls for careful design techniques. Operational amplifiers for direct-analogy computers are chopper-stabilized d-c amplifiers designed for use from direct current to 1,000 cps. Typical performance of such an amplifier is given by the following:

Drift stability: 50 μv per hr (referred to input)
Output voltage: $\pm$100 volts
Output current: $\pm$30 ma
Open-loop gain: over 1.5×10^7 at direct current, over 1.5×10^4 direct current to 400 cps
Characteristics with gain of 100:
Phase shift under 1 deg at 1,000 cps
Response down less than 3 db at 70,000 cps
Output impedance less than 2 ohms at 100 cps and less than 4 ohms at 1,000 cps
Hum and noise less than 100 μv (rms) referred to input

9.2.8. Multipliers and Function Generators. Direct-analogy computers employ conventional multipliers and function generators designed to use voltages as their inputs, and to produce voltages as their outputs. In the direct-analogy computer, then, these components must be employed in circuitry where only the voltages have significance, even if the simulation concerns terms which are actually represented in the passive-element network by currents. For example, if it is desired to produce a current at one node which is an arbitrary (nonlinear) function of the voltage at another node, the circuit of Fig. 9.2.9 will serve, if the transfer function $F(p)$ is replaced by a function generator. For use in computers with an upper frequency limit of 1,000 cps or over, high-speed electronic multipliers of recent commercial design are satisfactory. To generate arbitrary functions of a dependent variable (voltage) photoformers and diode networks are satisfactory devices. These function generators, as well as simpler switching devices, may also be used to generate functions of time (see also Art. 9.2.10).

For low-frequency direct-analogy computers based on resistor networks and resistor-capacitor networks, it is possible to use slower components. Servomultipliers are commonly used in such installations, and mechanical curve followers serve well as function generators.

9.2.9. Special Nonlinear Elements. The engineer using the direct-analogy computer tends to identify the various parts of the computer circuit with the individual components of the physical system he is simulating, rather than with abstract terms in mathematical equations. In using function generators and other active elements to simulate nonlinear effects it is desirable to continue this identification and to consider a given portion of the computer circuitry as representing a "nonlinear spring" or "nonlinear damping force," rather than just simulating a nonlinear term in an equation. The following are a few examples of the simulation of nonlinear elements. They are all based on a dynamic modal analogy for mechanical systems, but the same techniques will apply to other analogies.

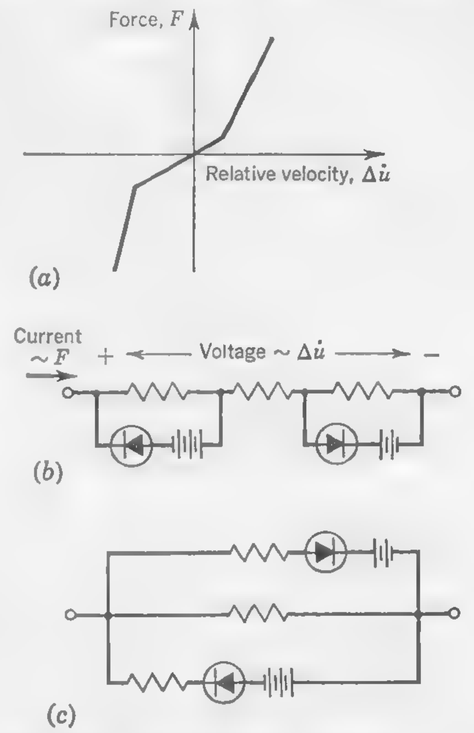


FIG. 9.2.13. Nonlinear damping force. (*a*) Desired force-velocity relation. (*b*) This circuit implements the characteristic by eliminating a series resistor at high current values. (*c*) This circuit implements it by inserting a parallel resistor at high voltages.

The simulation of a nonlinear damping force, in which the force depends only on the relative velocity of the points to which it is connected, is not very difficult. Figure 9.2.13 shows a force-velocity characteristic made up of several straight-line segments. Two-terminal networks are given which will simulate this nonlinear damper. Each of these networks is composed of resistors, diodes, and batteries.

The simulation of a nonlinear spring, or a nonlinear mass, by nonlinear inductors or capacitors is not so simple a matter. This is partly because the concepts of variable spring constant and variable mass are not always inherently very clear.

As an illustration of the sort of ambiguity which may be encountered, consider a capacitor which represents a variable mass in motion. Assume now the capacitor, for some reason, is to be doubled by the addition of a second capacitor of equal magnitude in parallel to it. If the second capacitor is initially uncharged, at the moment of switching the combined capacitor will retain the same electrical charge which the single capacitor had originally, but the voltage will be reduced to one-half its original

value, and one-half the stored energy will have been dissipated. The mechanical analogy of this process would be for the moving mass to crash into and absorb a stationary blob of putty with a mass equal to itself. Now consider the reverse of this process. The capacitor is to be restored suddenly to its original value by separation of the added parallel capacitance. Electrically, no energy is lost, although half the energy is dispatched with the separated capacitor. In this instance the voltage across the computer capacitor remains constant. The mechanical process analogous to this event would be the separation of the blob of putty from the original mass. However, the velocity of neither the original mass nor the added mass is changed; so the blob of

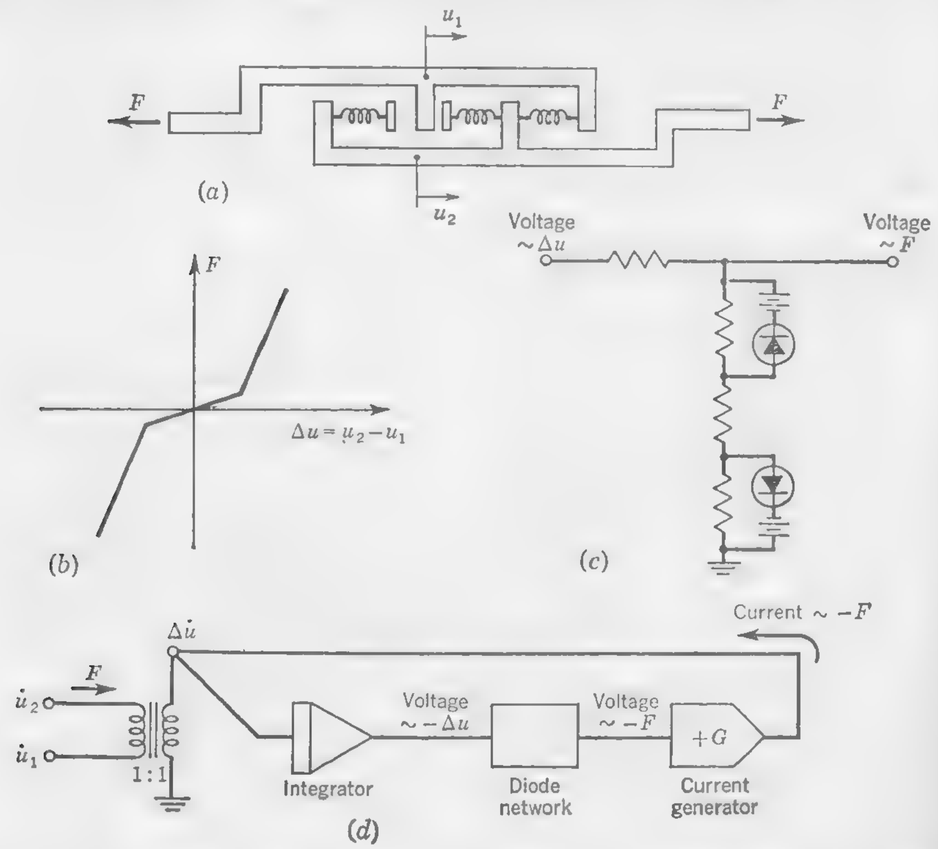


FIG. 9.2.14. Nonlinear spring. (*a*) Spring configuration. Spring constant is increased at large deflections. (*b*) Force-displacement characteristic. (*c*) Diode network simulates nonlinear characteristics. (*d*) Direct-analogy computer circuit. In the passive-element network, $\dot{u}_1$ and $\dot{u}_2$ are voltages at the nodes which simulate those coordinates. The "diode network" has the circuit of (*c*) or, in more complex cases, would be replaced by a complete function generator.

putty is permitted to tag along behind the original mass. The ambiguity is clear. By switching in additional capacitance, we (mechanically) conserve momentum but lose velocity. By disconnecting the additional capacitance we conserve velocity but lose half the momentum. Evidently, in discussing a nonlinear, or variable, mass, it is necessary to specify the required changes in energy or momentum.

Similar considerations apply to descriptions of nonlinear springs. A nonlinear spring may usually be adequately described by a force-displacement curve, because the potential energy of the deformed spring is implicitly given by the area under such a curve. Figure 9.2.14 shows the steps which might be followed in the simulation of a sample nonlinear spring. In the dynamic nodal analogy the circuit voltage represents velocity. A simulation of the nonlinear spring uses a transformer to generate a

relative deflection-velocity coordinate. (In essence, this transformer merely reduces the simulation from a two-terminal spring to a one-terminal spring.) An integrator is next employed to obtain a voltage analogous to the deformation of the spring. After a diode network or other function generator correctly simulates the desired nonlinear dependence of the force on the deformation, a current generator is employed to insert into the passive-element circuit a current proportional to this force (Fig. 9.2.14*d*). If the diode network were removed and the integrator were connected directly to the current generator, this circuit, as expected, would behave just as an inductor would if connected between nodes $\dot{u}_2$ and $\dot{u}_1$.

A circuit employing a multiplier and amplifier can be used to simulate a *time-varying* passive element. Permitting coefficients to be functions of time does not make an equation nonlinear, and the establishment of time-varying passive elements does not make a nonlinear circuit. However, many of the same precepts apply. In particular, in establishing time-varying springs and masses, one must guard against all the ambiguities discussed above.

A general approach to the establishment of time-varying passive elements is based on the circuit of Fig. 9.2.12. The amplifier marked A is replaced by a multiplier. One of the inputs to the multiplier comes from the amplifier with a gain of 1, and the other multiplier input is an appropriate function of time $A(t)$, which may be derived from a function generator. It is evident that the impedance exhibited by this circuit is now $[1 - A(t)]Z$. By proper choice of $A(t)$ and the impedance Z, any impedance may be programmed.

PERIPHERAL EQUIPMENT

9.2.10. Excitation. The *excitation system* must provide proper *forcing functions*, *initial conditions*, and constant or time-variable *boundary conditions*.

In both transient and steady-state analyses, the excitation may consist of the specification of voltages at the nodes of the analog circuit or of currents which are inserted into the nodes of the analog circuit. Of course, in many instances the specified voltages or currents are identically zero. This condition is called a *constraint*, and it is most expediently carried out by grounding the appropriate nodes or opening the appropriate branches at the plugboards which establish the analog circuit. In this case, no recourse to any special excitation equipment is necessary. An example of this technique is found extensively in the analogies for mechanical systems. In the nodal analogy, free edges of the mechanical system are simulated by open circuits, and clamped edges of the mechanical system are simulated by short circuits to ground.

In a *transient analysis* it is necessary that the timing of all events be carefully synchronized and controlled. The events which must be synchronized include the following:

1. The establishment of initial conditions prior to the start of the solution
2. The generation and application of excitation voltages and currents
3. Synchronization of output displays, such as oscilloscope sweeps
4. Termination of the solution and restoration of initial conditions

In the most common systems for achieving the necessary synchronization of these events the key component is the *synchronous switch*. Figure 9.2.15 gives a typical synchronous-switch circuit. The synchronous switch itself consists of a fast-acting mercury relay. The relay is energized by the current flowing in a thyratron circuit. A very linear sawtooth wave is supplied to each synchronous-switch circuit, and this constitutes the timing pulse. For each synchronous switch the time delay from the beginning of the sawtooth wave until the first thyratron fires and the mercury relay is energized is controlled by a potentiometer. A second potentiometer controls the time interval from the start of the sawtooth wave until the mercury relay is deenergized.

In the computer installation there may be some ten synchronous switches. All are operated by a common sawtooth wave to ensure synchronism. The sawtooth-wave generator may be controlled so that it will produce just one sawtooth wave each time the operator activates it, or it may be permitted to produce repeated sawtooth waves continuously. These types of control would correspond to *one-shot* and *repetitive* oper-

ation of the entire computer. Also, by varying the rate of rise of the sawtooth wave, the operator can speed up or slow down all functions controlled by the synchronous switches. Synchronous switches employing basically the above circuitry and mercury relays are capable of maintaining specified time intervals to a precision of 0.1 msec in repeated operations.

Figure 9.2.16 indicates a typical use of the synchronous switches in the solution of a transient problem on the direct-analogy computer. In this example three synchronous switches are employed. One switch initiates the sweep in an oscilloscope on which the solution is to be displayed as a function of time. A second synchronous switch, by controlling the energizing and deenergizing of *slave switches*, establishes initial conditions up to the beginning of the solution time and reestablishes the initial

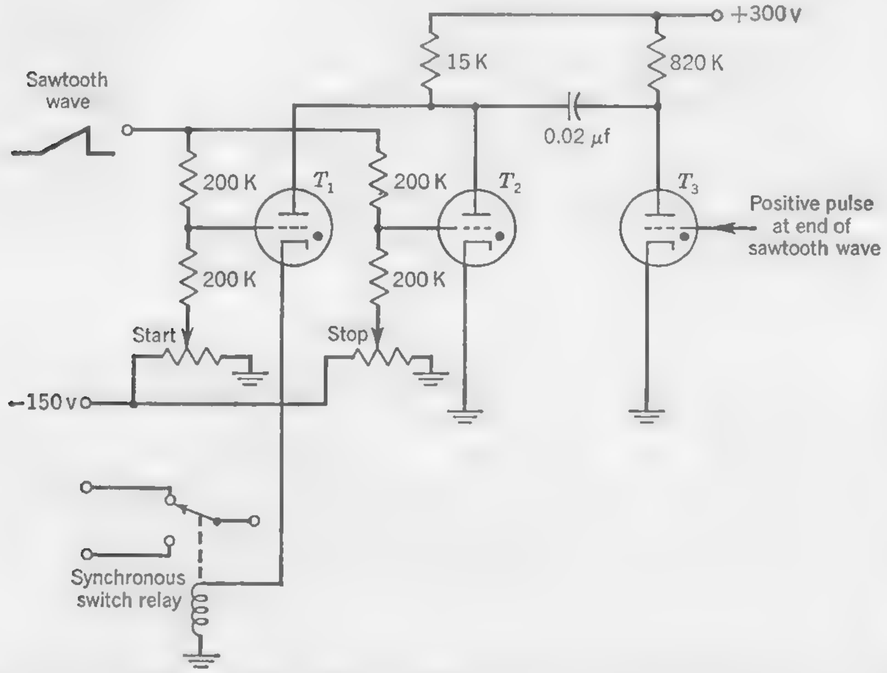


FIG. 9.2.15. Synchronous-switch circuit. After the start of the sawtooth wave, and delayed by a time interval which is determined by the start potentiometer, thyratron T_1 fires, energizing the relay. At a time determined by the stop potentiometer thyratron T_2 fires, extinguishing T_1 and deenergizing the relay. At the end of the sawtooth wave, a pulse fires thyratron T_3. This extinguishes T_2. T_3 is then extinguished by its own plate circuit.

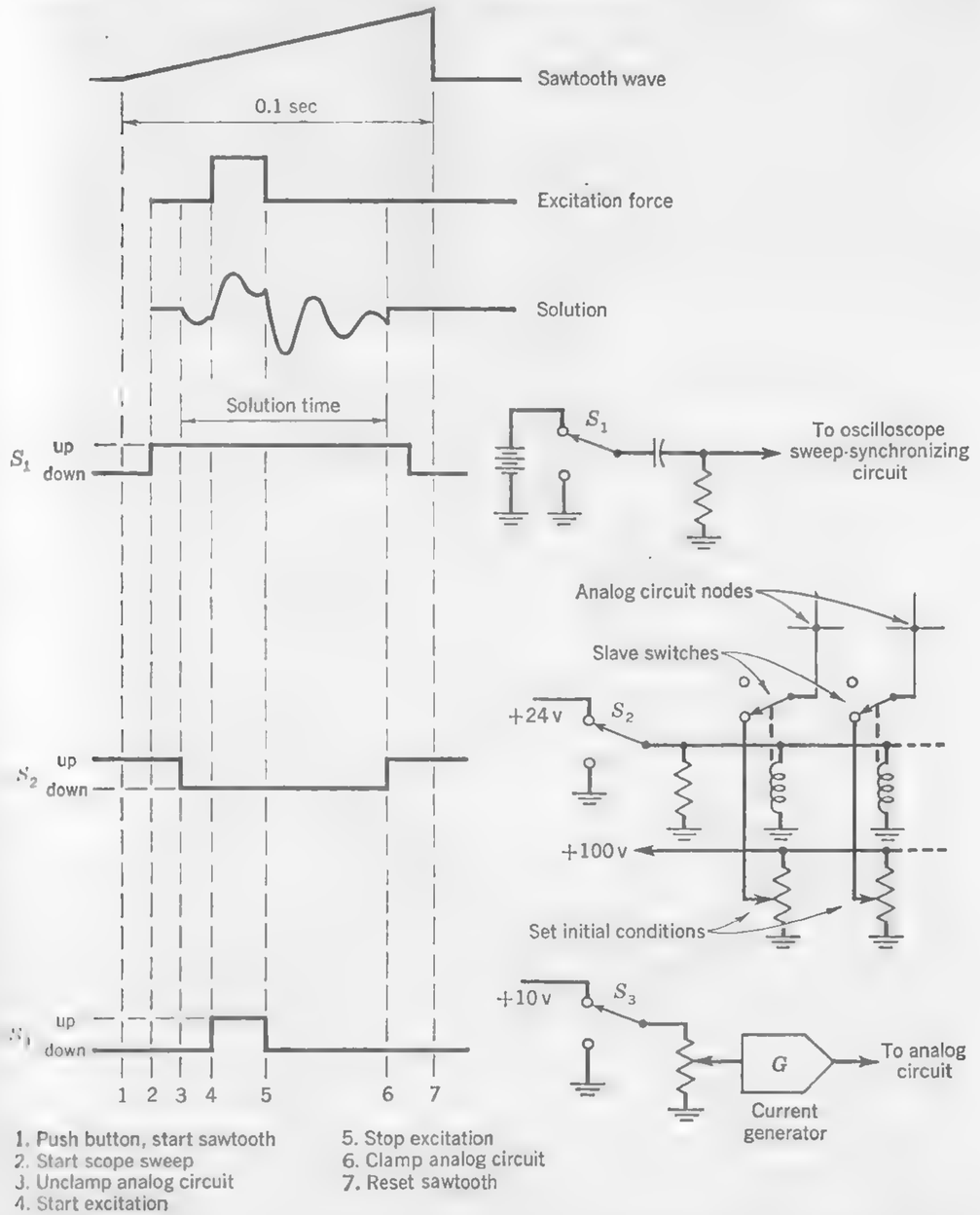
conditions after the end of the solution time. A third synchronous switch generates a forcing function, which in this case is a square current pulse applied during a part of the solution time.

In other problems the generation of more complicated excitation functions might make use of several synchronous switches. To generate certain types of functions, a d-c voltage supply and a synchronous switch may be used with other components as indicated:

Ramp function	Integrator
Sinusoidal pulse	Oscillator (d-c supply not required)
Exponential wave or damped sinusoid	Passive elements
Arbitrary function	Function generator

The final component in the generation of forcing functions will be a voltage amplifier or current generator, depending on whether a voltage or current is being specified.

In the *steady-state operation* of computers either d-c or a-c excitation may be required in the case of all-resistor networks, and a-c excitation will be required in the case of



1. Push button, start sawtooth
2. Start scope sweep
3. Unclamp analog circuit
4. Start excitation
5. Stop excitation
6. Clamp analog circuit
7. Reset sawtooth

FIG. 9.2.16. Use of the synchronous switches. This example shows the use of the synchronous switches in controlling the timing of events during the solution. All the synchronous switches are controlled by the sawtooth wave. In this example the sawtooth wave is initiated by a push button. (In repetitive-computer use the sawtooth automatically repeats continuously.) Synchronous switch S_1 initiates the sweep at the display scope. S_1 actuates the slave switches, which establish initial conditions at the beginning of the solution time and clamp the analog circuit at the end. S_3 is used here to generate the forcing function, a square-pulse current input.

other passive-element networks. In an all-resistor network d-c voltage excitation is best established by means of a d-c power supply and low-impedance voltage-dividing networks. A d-c excitation may be created by the use of a high-voltage d-c source and a fairly large series resistance. If the series resistance is not sufficiently large compared with the analog-circuit input impedance, the actual current which flows into

the analog circuit may differ somewhat from the desired current. It may then be necessary to perform an iteration, in which all such series resistances are reset.

Steady-state a-c excitation is used in the following examples of direct-analogy computing:

1. Determining the normal-mode characteristics of dynamic elastic systems
2. Determining the frequency response or transfer functions of systems
3. Determining stress distributions or influence coefficients of static elastic systems.

In many of these problems an audio oscillator and a voltage amplifier or current generator (see Art. 9.2.7) will be adequate. In some instances, however, it is necessary to generate simultaneously a number of a-c excitations, each to be inserted at a different node in the passive-element network. In particular this form of excitation is

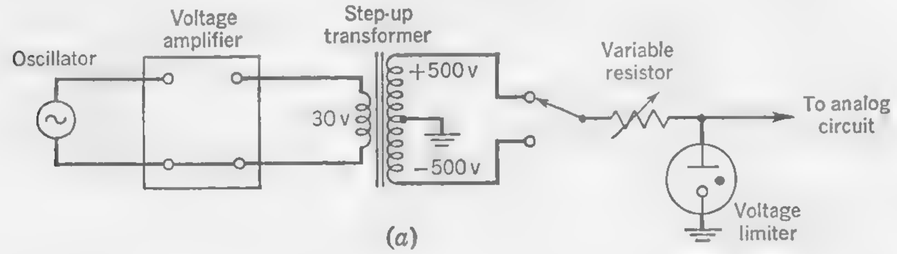


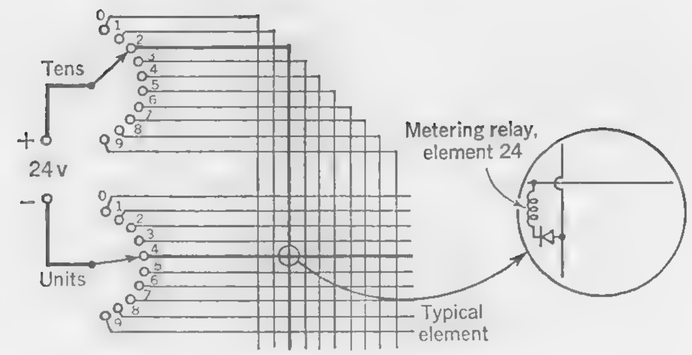
FIG. 9.2.17. Static current generator. (*a*) Basic circuit for generation of specified current at a fixed frequency. In supplying currents of under ±5 ma, output impedance (determined by setting of variable resistor) is over 100,000 ohms. In practice, one step-up transformer supplies six variable-resistor circuits, one voltage amplifier supplies four step-up transformers, and one oscillator supplies all required voltage amplifiers. (*b*) A refinement in which a one-to-one transformer subtracts out the analog-circuit node voltage. Thus the full 500 volts is across the variable resistor; so the effective output impedance is much greater than 100,000 ohms.

frequently required in static-stress analyses, in which it is required to simulate distributed, externally applied forces.

A simple circuit for an a-c generator is shown in Fig. 9.2.17*a*. This circuit consists essentially of a large a-c voltage source and a large series resistor. Figure 9.2.17*b* illustrates a refinement of this current generator in which the output impedance is increased to a more satisfactory value. In this circuit a cathode follower may be used for the amplifier with a gain of +1. A further refinement of the a-c generator is to construct the variable resistor of parallel-decade fixed resistors. By this means the current may be established by a decimal relation of the resistor setting switches, rather than by a reciprocal relation which is obtained with the usual series-decade resistors. In a typical computer installation 72 current generators make use of 72 variable series resistors, 12 step-up transformers, 3 voltage amplifiers, and a single oscillator.

9.2.11. Metering. The metering system is naturally divided into the following functions:

1. Element selection
2. Function selection
3. Display
4. Recording



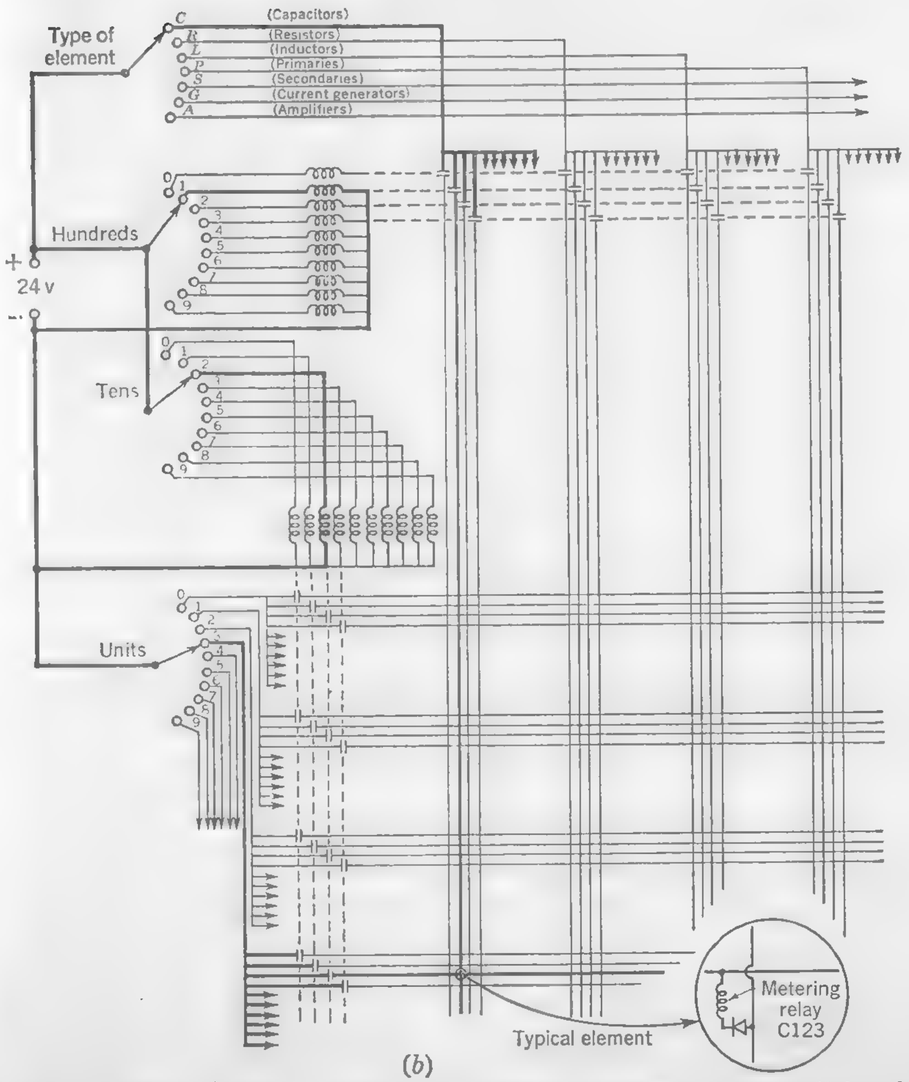


FIG. 9.2.18. Element-selector circuit. (*a*) Two-digit address. (*b*) Four-digit address. The selector units shown as rotary switches are actually push-button assemblies. For the element selected in each circuit as shown, only the heavy lines are active. The diodes permit current to flow from vertical lines to horizontal lines at only the one element-metering relay being selected. Not all the lines which exist are shown in (*b*).

The operation of the metering system requires different display and recording equipment for transient and steady-state operation, but the remainder of the metering system is applicable for both types of operation. Metering should affect computing voltages and currents as little as possible.

Associated with each passive element and with each computer amplifier is a *metering relay*. The purpose of the *element-selector* circuit is to permit the operator to select an element and to energize only the metering relay of that element. A cross-wire circuit is most commonly used for element selection in direct-analogy computers. Figure 9.2.18*a* gives a simple circuit for the selection of elements which are identified by a two-digit address. A more complicated circuit, utilizing multipole relays capable of handling the more usual requirement of a four-digit address system, is shown in Fig. 9.2.18*b*. In the application of this circuit, the cross wires and metering relays are

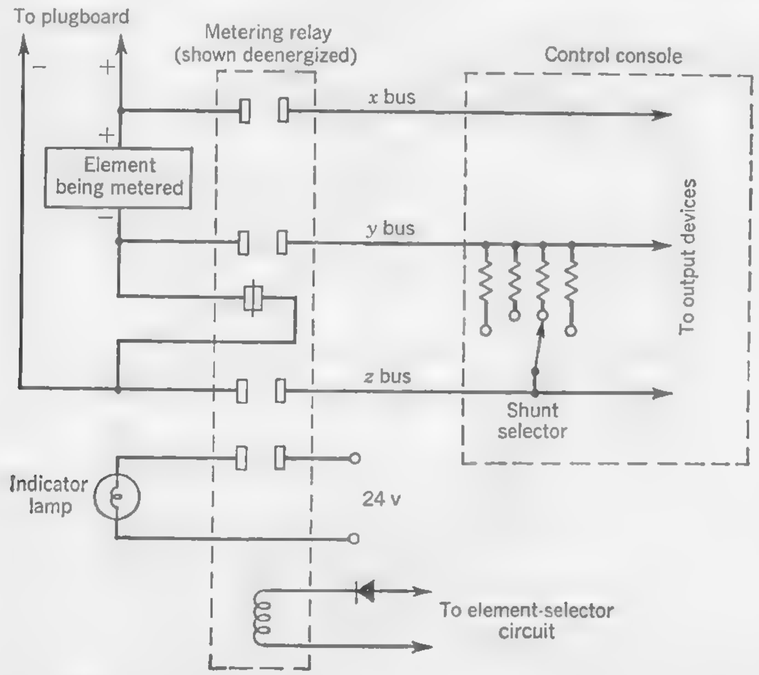


FIG. 9.2.19. Metering of elements When an element is selected, its metering relay is energized. Without disturbing the element's connection to the analog circuit, the relay makes available for measurement at the control desk the voltages and current associated with it. Only one element at a time is connected to the *xyz* metering busses.

located adjacent to the corresponding computer elements. The power supply, selector switches, and relays are located at the control desk. The selector switches are controlled by the operator by means of a push-button assembly.

Figure 9.2.19 shows how the energizing of an element-metering relay makes available at the control desk the voltages which exist at the terminals of the element and the current flowing through it. The x and y metering busses bring to the control desk the voltages which exist at the plus and minus terminals of the element, respectively. The current flowing out of the minus end of the element is routed through the y and z metering busses to the control desk, where a shunt is inserted for current measurement. [Although the two-terminal passive elements have no inherent polarity, they have plus and minus terminals when installed in the computer. The functions of this assignment of polarity are (1) to identify the terminals at the plugboard, (2) to specify adequately the sign of current measurements, and (3) in the case of transformers, to indicate the electrical polarity characteristics.] It is noted that only two-terminal elements conform to the circuitry shown on the metering relay. Transformers are

accommodated in this system by having one metering relay for the primary winding, and a second metering relay for the secondary. A computer amplifier is a three-terminal element, but measurements are made only at the output terminal.

Before the information on the x, y, and z metering busses is fed into the output devices, the operator performs a *function selection* as follows:

1. To measure the voltage at the plus terminal of the element select x ground.
2. To measure the voltage at the minus terminal of the element select y ground.
3. To measure the voltage across the element select x-y.
4. To measure the current flowing through the element select y-z.

When current is being measured, the proper metering shunt resistance must be connected between the y and z busses. This connection is established at the control console. In addition to the simplified circuitry shown in Fig. 9.2.19, it is frequently necessary to interpose additional components between the element being metered and the output devices to avoid disturbing the analog circuit. These additional components include potential transformers, current transformers, and d-c amplifiers.

Table 9.2.3. Output Devices

	Type of signals accommodated		
	Transient	Steady-state	
		A-C	D-C
Display devices	Cathode-ray oscilloscope Strip-chart recorder*	Vacuum-tube voltmeter Digital voltmeter† Cathode-ray oscilloscope Resolver	Vacuum-tube voltmeter Digital voltmeter
Recording devices	Camera which photographs oscilloscope Strip-chart recorder*	Printer operated from digital voltmeter	Printer operated from digital voltmeter

* Use is restricted to low-frequency computers.
† Provision must be made for operating the digital voltmeter from a-c voltages.

Table 9.2.3 summarizes various *output devices* employed in direct-analogy computers. Note that such machines do not admit a HOLD condition, because of the low-frequency characteristics of transformers and inductors.

In the measurement of steady-state a-c quantities, it is often necessary to measure not only the *amplitude* of the quantity, but also its *phase angle* with respect to some reference. An accurate and convenient method of phase measurement is shown in Fig. 9.2.20. This circuit employs resolvers as devices which produce sinusoidal output with amplitude and frequency determined by the input supply, and with phase angle (relative to a reference) determined by the orientation of the rotor.

When a mechanical printer is connected to record steady-state voltages and currents, the information concerning the magnitude of these quantities is supplied by the digital voltmeter. Simultaneously, information giving the identity of the element being metered is supplied by additional contacts on the element-selector switches, and information giving the type of quantity measured (voltage or current) is supplied to the printer from the function-selector switch. Each time the printer prints, a record is created showing what element was measured, what type of electrical quantity was measured, and the magnitude of the measurement which was made.

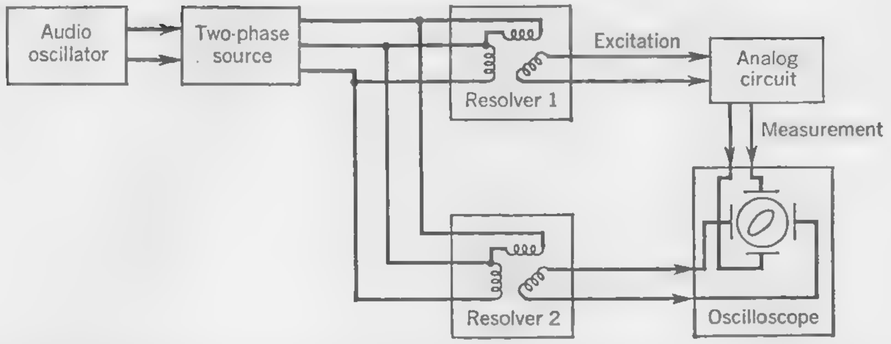


FIG. 9.2.20. Use of resolver. The two-phase source supplies to the resolver stators a-c voltages 90 deg apart in phase. The resolver output is fixed in amplitude, and its phase depends on rotor orientation, which is read from a dial. Resolver 1 is set to 0 deg, and resolver 2 is adjusted until the oscilloscope indicates no phase difference between its inputs. Resolver 2 gives the phase shift of the analog circuit. Accuracy of 1 deg is obtainable without calibration from 100 to 1,000 cps.

Table 9.2.4. Location of Direct-analogy Computer Components

Class of element	Location	
	Control console	Analog circuit units
Passive analog elements		Transformers Inductors Resistors Capacitors
Active analog* elements		Amplifiers Multipliers Function generators
Connections	Control console plugboard Buses	Element plugboards Buses
Excitation	Sawtooth wave generator Synchronous switches Voltage amplifiers Audio oscillator D-C voltage source	A-C generators Slave switches Voltage amplifiers
Metering	Element-selector switch Shunt-selector switch Function-selector switch All output devices	Metering relays Metering buses
Power supplies	Power supplies for all excitation and metering equipment	Power supplies for active analog elements

* In many installations the active analog elements are not segregated from the passive-element units.

9.2.12. Control Console. The control console houses those components of the excitation and metering systems which should be readily accessible to the operator and which need not be located in close proximity to the passive and active analog elements. Table 9.2.4 summarizes this by tabulating the major components according to their location in the computer.

OPERATION

9.2.13. Operating Features. The most striking characteristic about the operation of the direct-analogy computer is that *the problems handled are large:* another is *speed of operation.* It is not at all uncommon for a passive-element network involving over 100 independent nodes to be set up in a direct-analogy-computer installation. In terms of a dynamic structural system, this corresponds to the representation of over 100 different independent deflections. In comparison with electronic-analog-computer operation it must be realized that the different computers are best suited for different sorts of problems; a problem of the size described might easily call for 600 to 800 operational amplifiers.

In a typical analysis a period of preliminary computation, culminating in a complete set of circuit diagrams and initial element settings, occupies some 2 to 4 weeks, depending on the complexity of the system to be analyzed. A period of 2 to 4 days may be employed in setting up the circuit on the analog computer and in making those checks necessary to ensure that the desired system has been simulated. The period of obtaining problem solutions may last anywhere from several days to a month, depending on the amount of data required, in the case of an analysis, or on the amount of work necessary to obtain a satisfactory configuration, in the case of a design problem.

9.2.14. Checking. The purpose of checking is to establish confidence that the simulation achieved by the computer corresponds to the desired system. Deviations from the desired behavior could come from several sources. The first source, and the easiest to check, is the *behavior of the individual passive elements* themselves. It is easy to use plugboards which have been wired specifically for testing the passive elements and, by means of the computer metering system, ascertain that each inductor is behaving like an inductor, that each transformer obeys the ideal transformer equations, etc., within the applicable performance specifications. The passive elements are stable and inherently free of erratic troubles, and unless one is damaged by exceeding the rated voltage or current level, little difficulty is expected here.

When the circuit for a given simulation is installed in the computer, the next check which is normally made is a *topological* one—a verification that each element is connected to the proper nodes. A direct check may be made on the connections at each node. All elements connected to a given node should have the same voltage at the proper terminal. A check can also be made of Kirchhoff's law regarding the sum of all currents flowing into a given node; this guards against the connection of undesired extraneous elements to the node.

A final check of the passive-element network is a *functional* one, and is generally based on the physical properties of the system being analyzed. For example, in a dynamic structural analysis, the following checks may be performed:

1. *A check of the rigid-body kinematics.* By temporarily simulating the system in a rigid, free-body state, all geometrical relations may be verified by simple voltage measurements at all transformer connections.
2. *Rigid-body dynamic relations.* The rigid-body inertia matrix characteristics may be verified directly by applying an a-c voltage, in turn, to each rigid-body degree of freedom, and measuring the currents which flow in the constraints imposed on other degrees of freedom.
3. *First vibration-mode frequency.* Based on an assumed or measured shape of the first vibration mode, simple analytical methods lead to a calculation of the first vibration-mode frequency. This may be checked against the frequency as measured on the computer.
4. *Orthogonality of normal modes.* A verification of the orthogonality of different normal modes of vibration of an elastic structure will act as a check on the mass characteristics and the geometrical representation.

The *checking of amplifiers and other active elements,* both as individual components and as portions of subsystems, is accomplished in a manner similar to that employed in electronic analog computers.

9.2.15. Accuracy. The following effects are the source of the principal limitations on the accuracy of the solution of problems by the direct-analogy computer.

1. Inability to describe the system to be analyzed with complete accuracy
2. Nonapplicability of the assumed laws governing the behavior of the system
3. *Lumping* of continuous structures
4. *Roundoff error* in setting decade elements or other elements with discrete settings
5. *Parasitic effects of the computer elements*
6. Parasitic effects of the metering system
7. System errors of the metering circuitry

The first two items are essentially outside the realm of a discussion on computers. The third item, lumping, is a major limitation on the accuracy of the solution of distributed systems by most methods of computation (see also Sec. 9.1). The fourth source of inaccuracy, roundoff error, is insignificant if the deviations from the desired settings are random in nature. However, if the passive elements have a system error, or if the roundoff error is always in the same direction, the error may approach the same order of magnitude as the tolerance which exists in the specifications for passive-element accuracies.

Table 9.2.5. Size of Direct-analogy Computer Installations

	Approx quantities	
	Smallest	Largest
Number of passive elements:		
Transformers	40	200
Inductors	30	150
Resistors	60	300
Capacitors	50	160
Number of active elements:		
Operational amplifiers*	40	170
Function generators	.....	10
Multipliers	.....	10
A-C generators for static-stress excitation	.....	90
Length of computer, ft	15	90
Cost of complete installation	Approx $100,000	Over $500,000

* May be connected as integrators, summers, current generators, or as general operational amplifiers.

The fifth and sixth items, limitations on accuracy due to element and metering parasites, play an important part in the design of the computer and also in the creation of analogies and selection of scale factors for a particular problem. These inaccuracies, also considered in Sec. 9.1, are nominally held to 1 per cent. The actual maximum values of the parasitic effects permitted in the computer installation are usually part of the performance specifications. Although the nominal 1 per cent figure is achieved when the operating range of the passive elements differs by a factor of 100 from the associated parasitic values, it is still necessary to maintain vigilance in the design of analog circuits to avoid possible cumulative parasitic effects. The nominal accuracy of the seventh item, the metering system, is also 1 per cent, although this figure will vary in accordance with the specific output devices employed. Digital-voltmeter accuracies exceed this specification, but most of the devices employed to record transient solutions are not quite capable of meeting the 1 per cent figure.

9.2.16. Typical Installations. The chief use of the direct-analogy computer has been in the field of analysis and design of complex elastic structural systems. Computer installations for this purpose vary in size over a range of four or five to one. Table 9.2.5 gives approximate figures for the smallest and largest direct-analogy computer installations for this purpose.

PART 3: POWER-SYSTEM NETWORK ANALYZERS

By R. P. Washburn

INTRODUCTION

9.3.1. Power-system Network Analyzers. One of the oldest computer types still in production is the *network analyzer* used to study power-system networks. Each circuit of the power system is represented by a corresponding circuit in the analyzer, with corresponding interconnections. The analyzer units are scaled exactly to the actual system. Instrumentation is provided to measure electrical quantities such as voltage at busses, currents in lines, power consumed by loads, and reactive flow at various points.

9.3.2. D-C and Direct-analog A-C Network Analyzers. The first network analyzers were *d-c network analyzers* operated from batteries. In order to represent quantities on the analyzer, the transmission network was assumed to contain only reactance elements. On systems where the ratio of resistance to reactance is approximately constant for all units, a resistance can represent total impedance, and very accurate solutions can be obtained. Even today, many utilities have a d-c board to obtain approximate solutions to their problems. The first *a-c network analyzer* was built in 1929 by a joint effort of the Massachusetts Institute of Technology and the General Electric Company; it was a 60-cycle design operating directly from the power line. This analyzer is still in use in Puerto Rico and has been modernized several times.

Because of the high cost of low-frequency reactors representing the inductive portion of the line and load impedances, the obvious conclusion is to use high audio frequencies; but metering problems, shielding, and stray capacitance effects limit the range of useful frequencies. General Electric adopted 480 cycles as an operating frequency; this is the eighth harmonic of 60 cycles and is easily generated by a synchronous-motor-generator set. Westinghouse chose the 440-cycle A pitch as their standard frequency.

A few boards have been made utilizing 10,000 cycles as the fundamental frequency to permit the use of modern high-Q audio-frequency inductors. The Iowa State College analyzer and the General Electric miniature network analyzer (1958) are examples of such "high-frequency" network analyzers.

9.3.3. Modified-analog Network Analyzers. The I.T.E. Circuit Breaker Company approached the reactor problem by using a 60-cycle frequency and the *5-deg phase-shift method of metering.* I.T.E. makes all their reactors with nominal losses and trims the losses by adding resistance until the phase angle between the applied voltage and the exciting current is exactly 85 deg. This allows the reactor designer to increase his copper and iron losses appreciably. If the metering circuits (wattmeter and varmeter) are offset by 5 deg, the I.T.E. reactors appear to have infinite Q, or no losses, thus facilitating problem setup and calculation. With such a metering circuit, a pure resistance load would read incorrectly: parallel capacitance must be incorporated to counter the 5-deg lag in the metering circuit. In order to represent pure capacitance, it is necessary to add special adjustments in each computer setup.

Analyzers employing the *conjugate-impedance method* substitute inductors for capacitors, and vice versa; the metering circuits are reversed to correct the representation.

The Portland General Electric Company has marketed a network analyzer which employs *separate circuits for in-phase and quadrature components.* These components are added by conventional wattmeters.

Modified-analog network analyzers sacrifice the direct-analog representation and may confuse the occasional user. Of the 75 analyzers known in 1958, 38 are Westinghouse 440-cycle units and 11 are General Electric 480-cycle units, both using conventional metering and direct-analog circuits.

DESIGN OF NETWORK ANALYZERS

9.3.4. Network-analyzer Components. Most network calculators are made to represent a large variety of systems rather than to solve a specific network. Table 9.3.1 lists the type and range of circuits furnished on the Westinghouse Corporation a-c network calculator located at East Pittsburgh. This is the largest analyzer presently manufactured. The average problem involves 12 to 18 *generators* with a proportional number of *line and load units.*

Table 9.3.1. Type and Range of Circuits Furnished on the Westinghouse Electric Corporation A-C Network Calculator at East Pittsburgh, Pa.

No.	Type of circuit	Range of adjustment quantities in per unit (Art. 9.3.5)	Steps of adjustment
36	Automatic generators	Voltage 0–3.4	Continuous (3 ranges)
		Phase angle 0–360	Continuous
36	Generator reactance	Reactance 0–0.998	0.002
		Reactance 1.00–4.99	0.01
120	Line impedance	Resistance 0–0.998	0.002
		Resistance 1.00–3.99	0.01
		Reactance 0–0.996	0.002
		Reactance 1.00–4.99	0.01
184	Pi line	Resistance and reactance, same as line unit	
		Capacitance 0.001–0.41	0.001
		Susceptance in each leg of pi	
108	Load impedance	Watts 0.02–2.00	Continuous
		Watts 2.0–200	Continuous
		Vars 0.01–100	Continuous
		Load adjuster 90–110%	1%
		Voltmeter 0.6–1.20	Continuous
		Voltmeter 1.20–2.40	Continuous
48	Capacitor (standard)	Susceptance 0–1.0	0.0025
8	Capacitor (large)	Susceptance 0.25–8.10	0.25
72	Autotransformer	Voltage 0.80–1.245	0.005
56	Mutual transformer	1:1	Fixed
16	Jumpers		
2	Instrument desks		
1	Recording desk		

The 60-cycle analyzers draw their power from the three-phase mains; 400- to 500-cycle systems require some form of motor-generator set to convert power to the correct frequency and three-phase operation. Five to ten kilowatts of three-phase power are required at these frequencies to supply losses and circuit power. For "high-frequency" analyzers it is more economical to use electronic-oscillator techniques and electronic phase-shifting methods. With appropriate scaling, only a few watts of audio power are required, so that a much more convenient generator package results.

Since 1948, the large analyzers have incorporated *generator metering instruments* with each generator unit. These indicate the voltage, watts, and vars being delivered by the unit and can greatly decrease the problem-setup time, as the operators can adjust the voltage and phase angle without having to read the master instruments.

Line units and load units represent the transmission lines and the larger loads on the system. Both use variable resistance and inductive-reactance units. Impedances are calibrated in per-unit-impedance (Art. 9.3.5), and the value is selected by knobs on the front of each unit. Setting accuracy is within 1 per cent of the maximum setting. Modern analyzers usually have autotransformers associated with the loads to maintain

constant voltage on the load units as the bus voltage varies. A suppressed-zero voltmeter associated with each autotransformer permits one to reset the autotransformer to the previously calibrated voltage.

Stray capacitance to ground of transmission lines is represented by calibrated *fixed capacitor elements.*

Generator reactance units are used in fault studies to represent transient reactances of the machines. Since the Q of modern machines is high, generator-reactance units require high-quality reactors.

Long transmission lines may be represented by special *long-line units* composed of a standard capacitor unit, a standard line unit, and terminated in another capacitor unit.

Mutual transformers are used with line units to represent impedances common to transmission lines located on the same tower or right of way.

9.3.5. Base Quantities and Scaling. An easily obtainable set of interrelated electrical quantities are referred to as the *analyzer base quantities.* Once the nominal operating analyzer voltage is chosen together with a nominal current flow, all other quantities are established. *Analyzer capacitance and inductance* values can be calculated when the operating frequency has been chosen. Two commonly used sets of *base values* are 50 volts, 50 ma, 1,000 ohms impedance, 480 cps; and 100 volts, 1.0 amp, 100 ohms impedance, 440 cps.

To represent the power system on the analyzer, the rating of a large generator may be chosen as the system *base kva.* Commonly used base kva ratings are 10,000, 50,000, and 100,000 kva. The most commonly used system voltage is chosen as system *base voltage. System base current* and *system base impedance* are thus determined.

All system quantities are expressed in per unit, or fractions and multiples, of the system base quantities. The *system base quantities* are represented on the analyzer by the *analyzer base quantities.* The following relationships are, thus, defined:

$$\text{Per unit watts (or vars)} = \frac{\text{kw (or rkw)}}{(\text{base kva})\ 10^3} \tag{9.3.1}$$

$$\text{Per unit ohms} = \frac{Z\ (\text{ohms}) \times (\text{base kva})}{(\text{base kv})^2 \times 10^3} \tag{9.3.2}$$

9.3.6. Design Details. Problems associated with producing good *reactors* for the various analyzer frequencies were discussed earlier. Constant inductance throughout the range of operating circuits is difficult to achieve in reactors. The large air gap required to avoid saturation necessitates larger magnetization forces and thus physically larger inductors. Large wire sizes are required to keep the copper losses down, especially near the air-gap areas where magnetic fringing occurs and introduces losses in the wire strands. A typical value of Q for 400- to 500-cps reactors is about 30. Special reactors used to represent the synchronous impedance of the generators have been produced with Q values of 100.

Each *generator unit* consists of a source of power and means for changing the voltage and its phase angle. Modern analyzers incorporate instruments to indicate the voltage, current, watts, and vars delivered by each generator unit. Figure 9.3.1 shows a simplified diagram of a generator unit. A power selsyn shifts the phase angle of each of the generator outputs and converts the three-phase power to single-phase. By proper design, the voltage regulation can be maintained within 1 per cent of nominal value, and the phase angle can be set within ½ deg. When the base power is insufficient to drive the generator instruments to the accuracy required, a current amplifier is used to amplify the voltage drop across a resistor in the current line.

Automatic generator-load-setter units are generator units incorporating automatic means to maintain a constant output within about 1 per cent. An adjustable input command is set and the output of the generator unit is continuously compared with the command signal. Any deviation from the command signal causes a servomechanism to rotate the phase-angle selsyn, which adjusts the power output of the generator unit such that the error signal is reduced to zero. The General Electric system utilizes the Hall effect as a power-to-voltage sensor. The Westinghouse system uses the gen-

erator wattmeter as a sensor, with an auxiliary coil to indicate the position of the wattmeter needle.

Figure 9.3.2 shows the simplified circuit of a generator unit utilizing 10,000 cycles as the base frequency. Here the common reference is an oscillator adjusted to exactly 10 kc, and electronic circuits shift the phase through plus and minus 45 deg with respect to the oscillator. These two output signals are applied to a conventional inverter circuit. The variable resistor and fixed capacitor form an RC phase shifter with the potentiometer for a phase-shifter control. The output voltage is approximately constant in amplitude and can be changed from 0 to well over 90 deg. A quadrant-selector switch reverses the polarity of both the input and output signal to produce the quadrant selection. Phase control varies the phase angle within the selected quadrant. A pair of cathode followers drive the output stage of the generator amplifier with class B operation. Because of the low driving impedance, the output of the generator amplifier is a very good sine wave; the generator unit is capable of producing well over 1 watt of power. A loading resistor is placed across the output to reduce the effect of capacitance and inductance loads. Since the system base kva of this analyzer is represented by $\frac{1}{10}$ watt, one generator unit may handle at least 10 per-unit loads.

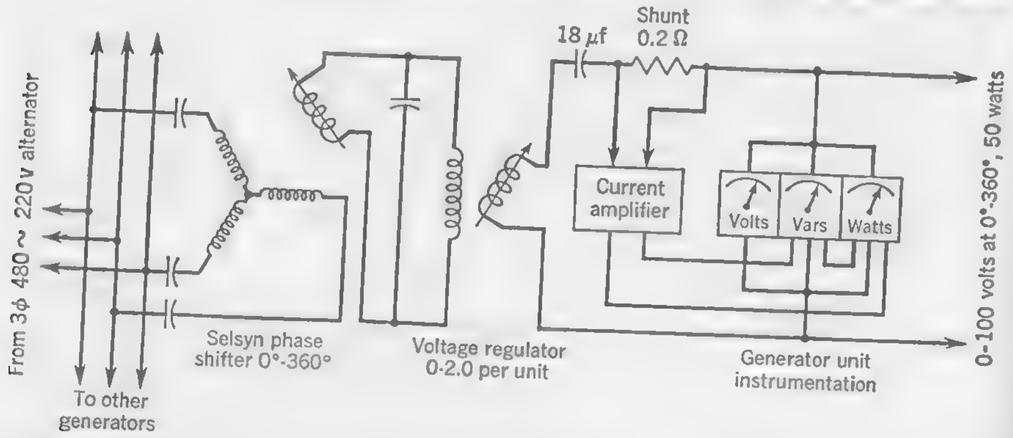


FIG. 9.3.1. Simplified circuit of 480-cycle generator unit showing the method of phase shift and voltage control and the unit instrumentation. (*General Electric Co.*)

Master instruments, which are used to read out answers, are similar to the generator instruments except that they are larger, more flexible, and are of ½ per cent accuracy compared with the nominal 1 per cent accuracy for the generator instruments. This instrumentation typically consists of a voltmeter, an ammeter, the watt-varmeter, or a wattmeter and a varmeter.

On 60-cycle analyzers, a conventional laboratory wattmeter, voltmeter, and ammeter is used. Special instruments are made for the 400- to 500-cycle analyzers. A desirable modification of these is the light-beam instrument, whereby a light beam is used to indicate on a long scale the instrument reading. These instruments have a high torque to reduce jewel roll and to provide fast response. This requires driving power, which is obtained from a current and a voltage amplifier each employing 40- to 60-db feedback.

Figure 9.3.3 shows the simplified circuit of a typical master instrumentation. Note that the voltage-amplifier input is obtained from the bottom of a large voltage-divider network compensated for the operating frequency; so that the output of the amplifier has correct phase relationship to the voltage input to the divider. Also note that there are two outputs of the attenuator, one in phase, the other exactly 90 deg lagging. This is to provide the circuitry for measurement of vars, which are defined as

$$EI \sin \alpha = \text{vars} \tag{9.3.3}$$

as compared with watts:

$$EI \cos \alpha = \text{watts} \tag{9.3.4}$$

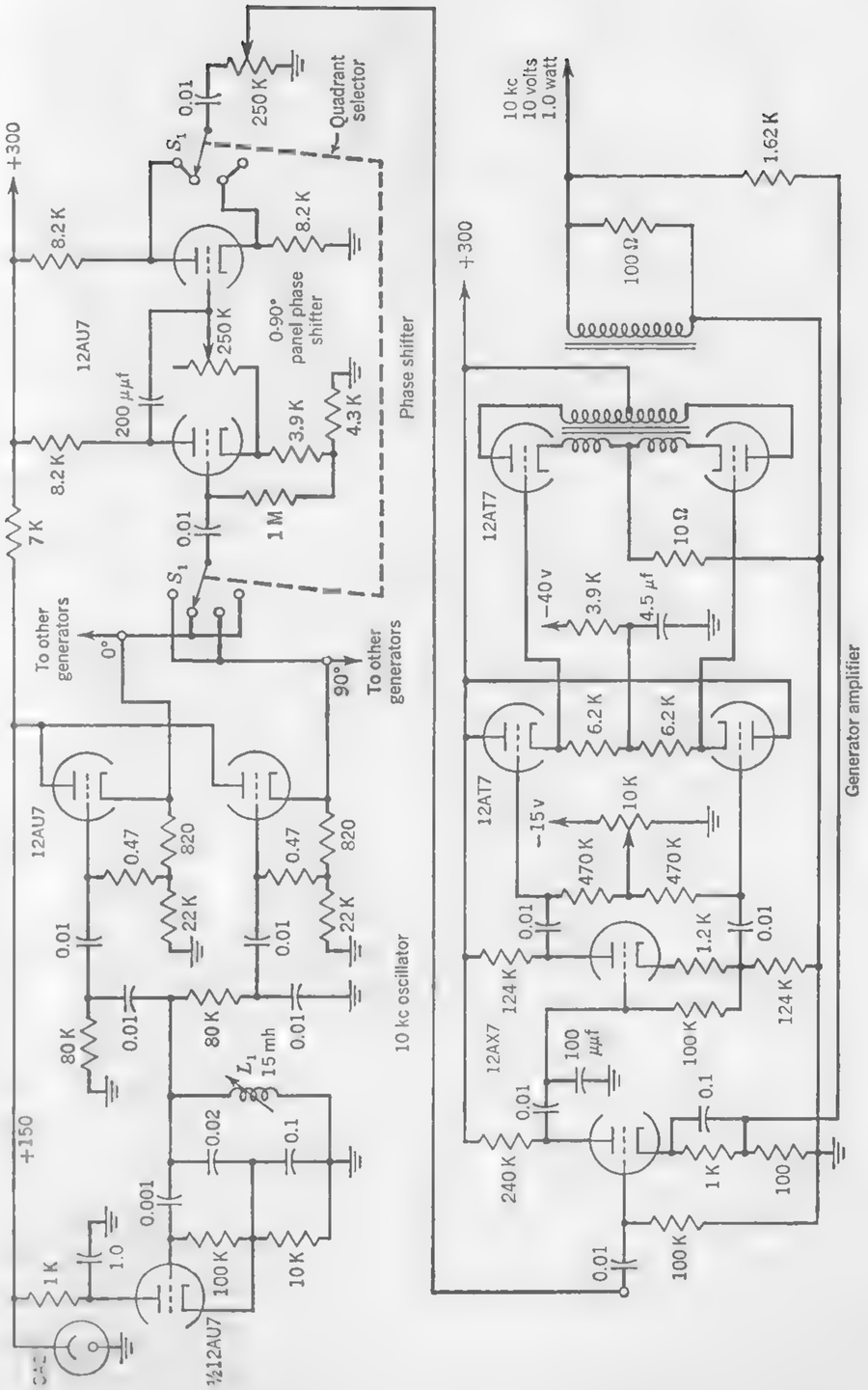


FIG. 9.3.2. 10-kc oscillator and generator unit. (*General Electric Co.*)

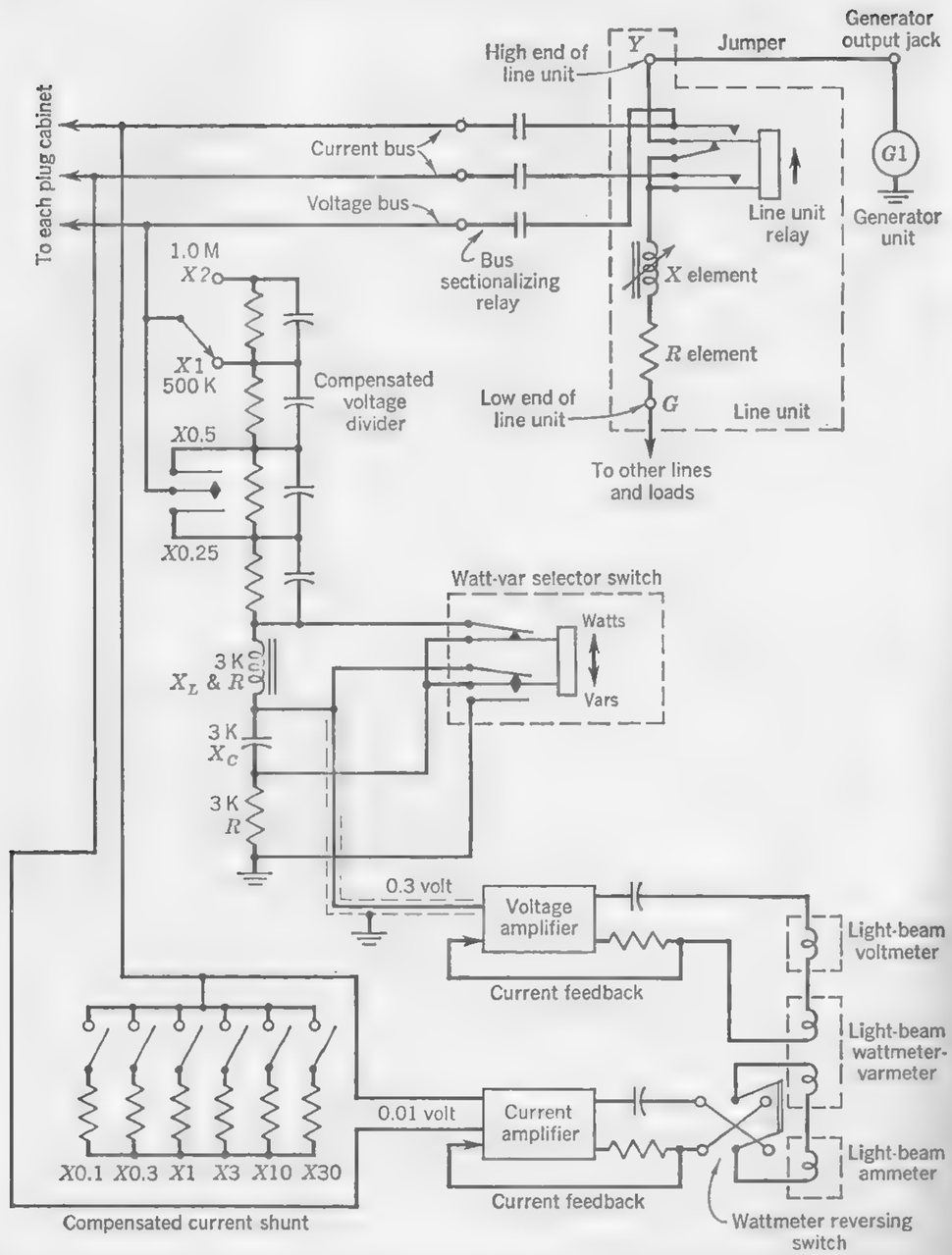


FIG. 9.3.3. Simplified circuit of the 480-cps master instrumentation and its connection to a line unit by means of the bus-sectionalizing and line-unit relays which are energized when this line unit is selected for metering. (*General Electric Co.*)

Substitution of a capacitor for the bottom resistor yields a 90-deg phase shift. In order to maintain current through the voltage divider exactly the same as with a resistor, a reactor is placed ahead of the capacitor. This reactor is wound to have the same resistance and reactance as the dropping resistor to maintain the voltage-divider ratio.

Circuit shielding introduces a shunting capacitance to ground and shifts the phase angle slightly. Thus the voltage-amplifier output is not exactly in phase with the

input voltage. In order to make the instrumentation read correctly, the current-source phase is shifted by the same amount. Thus, the current-measuring shunt is wound inductively, and provision is made to insert special iron in the core to tune its angle exactly with the voltage-amplifier output. Each of the various shunts is wound to exactly the same phase angle.

In a power system, the wattmeters used are generally made to read correctly on circuits having power factors of 90 per cent lagging to 90 per cent leading, since this is the range of loads generally encountered. On the analyzer the wattmeter is called upon to read any power-factor load from zero lagging to zero leading. Again, the same instrument is generally used as a varmeter and encounters a similar spread of phase angles. Thus special design and calibration are required to correct the instrument for other than normal power factors.

9.3.7. Semiautomatic Programming and Read-out. The following features are desirable to reduce the time required for problem setup and read-out of answers:

1. *Keyboard-actuated relay interconnections and metering*
2. *"Plotting tables"* which show circuit diagrams with lights indicating the point being metered
3. *Automatic scale changing* (see also Sec. 4.2)
4. *Reference-unit selsyns* providing reference voltages of known voltage and phase for comparison purposes

Digital read-out with automatic scanning would be another desirable feature.

SOLUTION OF NETWORK PROBLEMS

9.3.8. Required Data. The given system interconnections and circuit values should include line-resistance and reactance values in ohms. The values of capacitance installed on the lines or available for switching at specified points must also be known. Autotransformer specifications should include taps available and desired tap settings together with transformer impedances. If short-circuit studies are desired, one must specify generator transient and subtransient reactances. The expected range of loads must also be known.

A copy of the system diagram is marked up with the appropriate values of reactances, resistances, and power. The scheduled generation of each generator for the hourly, or minimum and maximum load conditions, must be given. The operator then selects line, load, and generator units according to the patching diagram and marks the corresponding analyzer numbers on the diagram. This becomes the main working sheet for the analyzer studies.

The analyzer operator and the customer engineers set the problem from this work sheet onto the analyzer by means of plugboards. The time required to set up a problem on the analyzer depends on the size of the problem, the amount of previous work done, and the familiarity of the customer with the analyzers, and may range from less than an hour to a number of hours.

The next phase of the study is the balancing of generators and loads to yield a *base-case solution* such that each generator supplies the specified voltage and power, and each load draws its assigned power. The procedure for this is to set each generator in succession to the assigned load, noting that, as each generator is adjusted, the previously adjusted generator flows will change. Automatic generator units provide automatic setting and balancing of each generator. This allows the operator to concentrate on the setting of the loads to the correct megawatts and megavar flow, and to regulate the associated load autotransformer so that the load being drawn is correct. Automatic load setters have been considered but are not in general use.

The base case is chosen to represent a known configuration of actual power flow and system performance so that the analyzer results may be compared with known data. After a rough check has been made to verify the analyzer results, a complete recording is made of the power flow. This base case will be the starting point for all future investigations and will be referred to again to isolate or verify future operation conditions.

9.3.9. Types of Problems Solved. The following table illustrates the utilization of a typical analyzer in terms of the type of problem and the approximate fraction of total computer time used for different studies.

Problem	*Approximate Time*
Load-flow studies	2/3
Fault studies	1/6
Stability studies	1/6
Relay studies and loss-formula studies	0+

As a result of a load study, the following quantities relative to the system are found:

1. Generator and bus voltages throughout the system.
2. Kilowatt, kilovar, and kva flow in any line, or to any associated circuit. Ratings of transformers and voltage regulators to produce designated system voltages can be determined.
3. Size of static or synchronous capacitors necessary for power-factor correction.
4. Study of changes in transmission voltages.
5. Effect of adding new loads to the existing system.
6. Effect of outages of certain pieces of equipment on the voltage and current readings of the system.

9.3.10. Fault Studies. Fault studies are conducted to determine fault currents and thus circuit-breaker ratings throughout the system. Since the fault current is many times the load-flow current, the analyzer voltage is generally reduced to one-half or one-quarter of that used for load-flow studies to permit metering the fault current.

During most fault studies, the loads, autotransformers, and capacitors need not be plugged into the analyzer. Large synchronous and nonsynchronous machine loads should be represented as generators in a fault study, since they act as generators contributing to the fault current. The impedance of the system generator units must be considered, this was not done in the load-flow studies. The value of impedance inserted in series with each generator unit is determined by the transient or the subtransient reactance of the generator. These are the reactances of the generators during the first few moments of fault conditions.

Since the basic network analyzer is a steady-state analog, the analyzer does not provide direct answers for the study of transient problems. The use of an iterative method permits use of network analyzers to compute transient characteristics of power systems. Such a transient study would require two or three times the time of a normal load-flow study, or from 2 to 3 hr compared with the 1 hr for an associated load-flow study.

References

1. Alternating Current Network Calculator Manual, *Technical Data 32-650A*, Westinghouse Electric Manufacturing Co., East Pittsburgh, Pa.
2. A.C. Network Calculator Operation, *Technical Data 58-1004*, Westinghouse Electric Manufacturing Co., East Pittsburgh, Pa.
3. G. E. Network Analyzer, GET-1285A, General Electric Company, Schenectady, N.Y.
4. Ryder, J. D., and W. B. Boast: A New Design for the A-C Network Analyzer, Paper 46-135, *Trans. AIEE*, **65**:674–677 (October, 1946). (10,000-cycle board at Iowa State College.)
5. Hoffman, C. H., and M. Lebenbaum: A Modern D-C Network Analyzer, *Elec. Eng.* AIEE Paper 56-20.
6. A New Principle Is Employed for 60-cycle A-C Network Analyzers, Paper 52-1[illegible] *Proc. AIEE*, **71** (1952).
7. Kuchni, H. P., and R. G. Lorraine: A New A-C Network Analyzer, *Trans. AIEE* **57**:67–73 (February, 1938).
8. Emns, Waldo E.: A Simple New Resistance Type A-C Load Flow Board, *AIEE Paper* 50-170, Pacific General Meeting, 1950.
9. Atherton, E. W.: Analyzer Features Made Possible by Zero-loss Reactors, I-T-E Circuit Breaker Company, Philadelphia, presented at American Power Conference, 1950.

PART 4: SPECIAL METHODS FOR THE SOLUTION OF FIELD PROBLEMS*

By WALTER J. KARPLUS

RESISTANCE NETWORKS

9.4.1. The Resistance-network Analogy. Purely resistive networks are particularly useful for the solution of Laplace's equation and Poisson's equation in one, two, or three dimensions in all the important coordinate systems. Under certain conditions their use may be extended to the simulation of fields governed by the diffusion equation, the wave equation, and the biharmonic equation as well. Although simple and straightforward in application, the use of resistance networks for the solution of field problems is a relatively recent development.[3,45] In general, the solution of field problems by means of resistance networks proceeds as follows:

1. The problem is formulated as a partial differential equation, and the field geometry, boundary conditions, initial conditions, internal excitations, and field parameters are specified.
2. The partial differential equation is approximated by a set of finite-difference equations. In effect the continuous field is replaced by a discretized equivalent, and attention is focused upon an array of discretely spaced points (nodes) in this field. The distance between adjacent nodes (net spacing) is selected so that the truncation errors resulting from the finite-difference approximation are sufficiently small.
3. A network of electrical resistors is now constructed such that each line segment of the finite-difference grid is replaced by an electrical resistor, and each node point becomes a junction point in the electrical network. The magnitudes of the resistors are determined from the finite-difference equations. An alternate physical approach, based upon the consideration of the field property represented by each resistor, leads to the correct resistance values without the use of finite-difference techniques.[1]
4. Voltage and current sources simulating the excitations are applied to the network at its external boundaries and at interior nodes, as required.
5. The voltage relative to an arbitrary reference point at each node of the network is noted and recorded. These voltages then correspond to the potentials existing at the corresponding points in the original field.
6. Equipotential lines, and possibly streamlines, are sketched by interpolation between points of known potential.

By using precision resistors and high-quality measuring instruments, one can obtain relatively high accuracies with this method. Accuracies to within 1 part in 10,000 are not uncommon.

9.4.2. Laplace's Equation in Rectangular Cartesian Coordinates. Laplace's equation

$$\nabla^2\phi = 0 \tag{9.4.1}$$

is expressed in cartesian coordinates in one, two, and three dimensions, respectively, as

$$\frac{\partial^2\phi}{\partial x^2} = 0 \tag{9.4.2a}$$

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \tag{9.4.2b}$$

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0 \tag{9.4.2c}$$

* The material of this section is largely based on W. J. Karplus, *Analog Simulation: Solution of Field Problems*, McGraw-Hill Book Company, Inc., New York, 1958.

The continuous field governed by these equations is now discretized as shown in Fig. 9.4.1. A coordinate grid is superimposed upon the field and attention is limited to the node points of this grid. A typical node (denoted by subscript 0) is selected, while adjacent nodes are labeled with subscripts 1, 2, 3, 4, 5, and 6. In accordance with

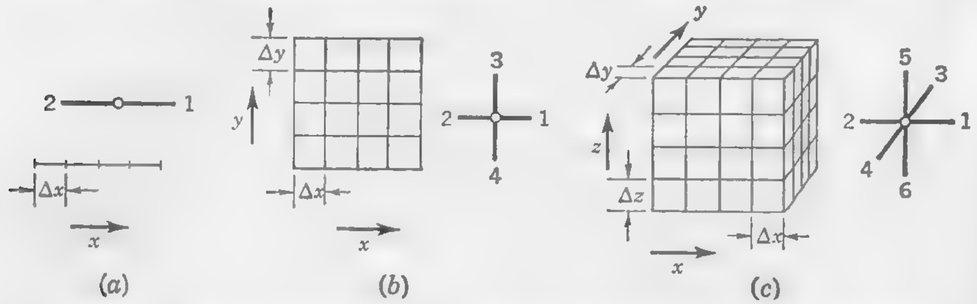


Fig. 9.4.1. Discretized one-, two-, three-dimensional fields with typical nodes.

well-known finite-difference techniques[1] the second-order space derivatives in Eq. (2) are approximated by "second-order central-difference" expressions according to

$$\frac{\partial^2\phi}{\partial x^2} \approx \frac{\phi_1 - \phi_0}{\Delta x^2} + \frac{\phi_2 - \phi_0}{\Delta x^2} \tag{9.4.3a}$$

$$\frac{\partial^2\phi}{\partial y^2} \approx \frac{\phi_3 - \phi_0}{\Delta y^2} + \frac{\phi_4 - \phi_0}{\Delta y^2} \tag{9.4.3b}$$

$$\frac{\partial^2\phi}{\partial z^2} \approx \frac{\phi_5 - \phi_0}{\Delta z^2} + \frac{\phi_6 - \phi_0}{\Delta z^2} \tag{9.4.3c}$$

Inserting these expressions in Eq. (2),

$$\frac{\phi_1 - \phi_0}{\Delta x^2} + \frac{\phi_2 - \phi_0}{\Delta x^2} = 0 \tag{9.4.4a}$$

$$\frac{\phi_1 - \phi_0}{\Delta x^2} + \frac{\phi_2 - \phi_0}{\Delta x^2} + \frac{\phi_3 - \phi_0}{\Delta y^2} + \frac{\phi_4 - \phi_0}{\Delta y^2} = 0 \tag{9.4.4b}$$

$$\frac{\phi_1 - \phi_0}{\Delta x^2} + \frac{\phi_2 - \phi_0}{\Delta x^2} + \frac{\phi_3 - \phi_0}{\Delta y^2} + \frac{\phi_4 - \phi_0}{\Delta y^2} + \frac{\phi_5 - \phi_0}{\Delta z^2} + \frac{\phi_6 - \phi_0}{\Delta z^2} = 0 \tag{9.4.4c}$$

Referring to Fig. 9.4.2, the sum of the currents into each typical node must equal zero. Therefore

$$\frac{V_1 - V_0}{R_1} + \frac{V_2 - V_0}{R_2} = 0 \tag{9.4.5a}$$

$$\frac{V_1 - V_0}{R_1} + \frac{V_2 - V_0}{R_2} + \frac{V_3 - V_0}{R_3} + \frac{V_4 - V_0}{R_4} = 0 \tag{9.4.5b}$$

$$\frac{V_1 - V_0}{R_1} + \frac{V_2 - V_0}{R_2} + \frac{V_3 - V_0}{R_3} + \frac{V_4 - V_0}{R_4} + \frac{V_5 - V_0}{R_5} + \frac{V_6 - V_0}{R_6} = 0 \tag{9.4.5c}$$

An exact analogy between field potential ϕ and electrical voltage V will exist if the resistors in Eq. (5) are proportional in magnitude to the squares of the corresponding net spacings in Eq. (4). To obtain convenient element values both sides of Eq. (5) may be multiplied by a scale factor R_0. Equations (4*a*), (4*b*), and (4*c*) can also be multiplied by Δx^2, $\Delta x\,\Delta y$, and $\Delta x\,\Delta y\,\Delta z$, respectively. Then

$$R_1 = R_2 = R_0 \tag{9.4.6a}$$

$$R_1 = R_2 = R_0\frac{\Delta x}{\Delta y} \qquad R_3 = R_4 = R_0\frac{\Delta x}{\Delta y} \tag{9.4.6b}$$

$$R_1 = R_2 = R_0\frac{\Delta x}{\Delta y\,\Delta z} \qquad R_3 = R_4 = R_0\frac{\Delta y}{\Delta x\,\Delta z} \qquad R_5 = R_6 = R_0\frac{\Delta z}{\Delta x\,\Delta y} \tag{9.4.6c}$$

The magnitudes of resistors along streamline boundaries of the field or in the immediate vicinity of boundaries which do not coincide with the grid lines of the finite-difference grid must be calculated separately. In the case of cartesian coordinates, resistors along streamline boundaries are in general twice as large as the corresponding interior resistors. Figure 9.4.3 illustrates a resistance-network analog for a two-dimensional field having two equipotential and two streamline boundaries. In practice, of course, a much finer network spacing would be used.

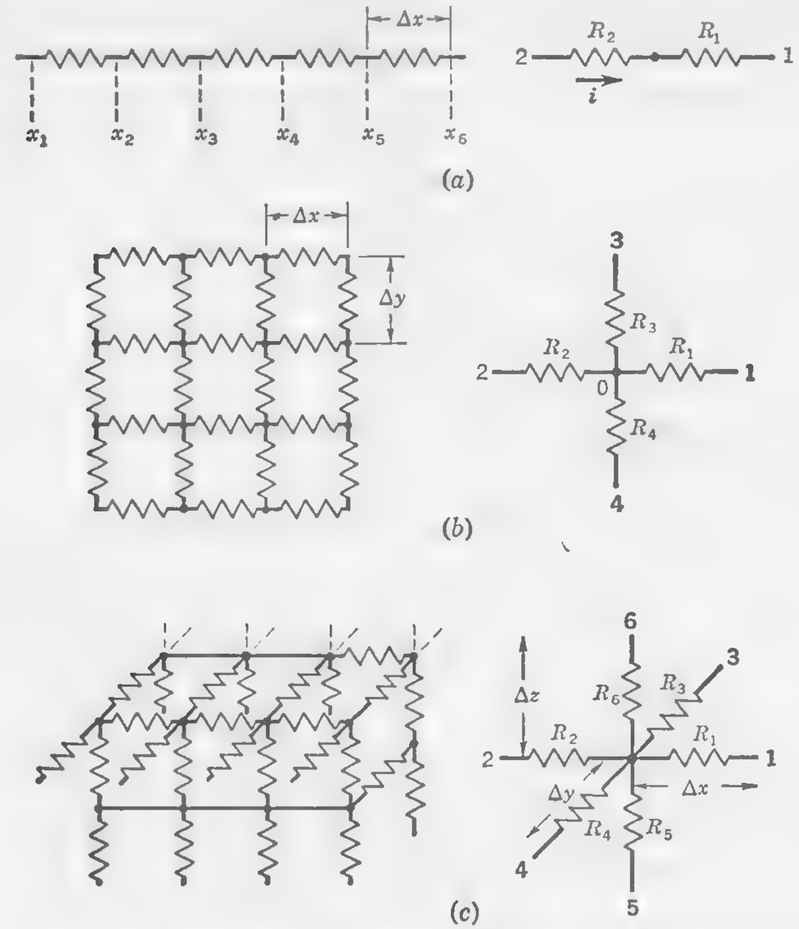


Fig. 9.4.2. One-, two-, and three-dimensional rectangular resistance networks and typical nodes.

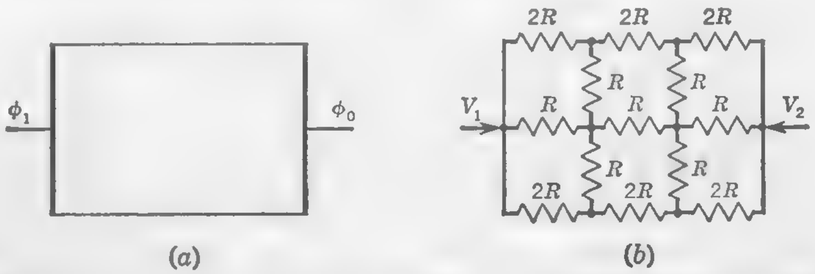
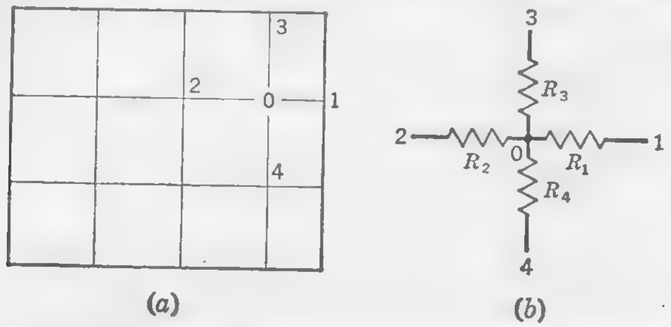


Fig. 9.4.3. (*a*) Uniform two-dimensional field. (*b*) Resistance-network analog.

If the boundaries of the field do not coincide with the node points of the discretized analog circuit, the resistors in the immediate vicinity of the boundaries are modified.

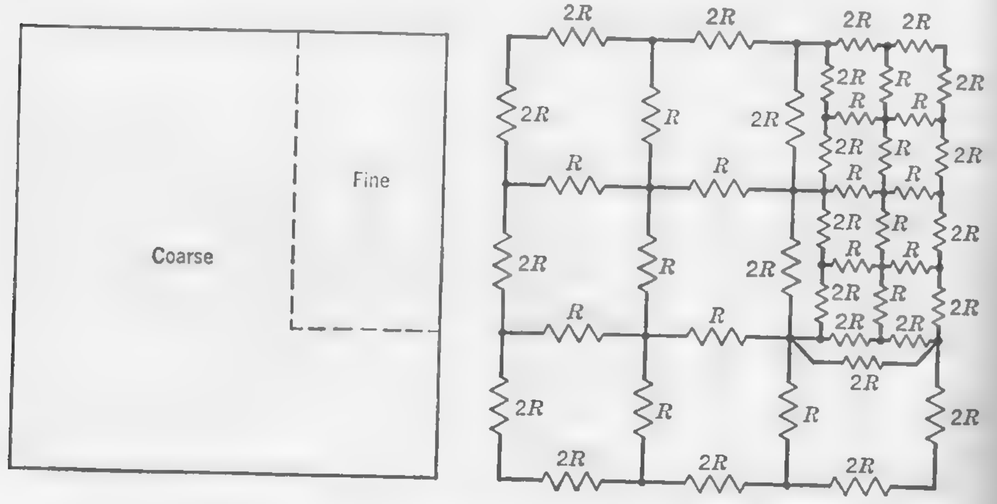
For example, in a field shown in Fig. 9.4.4*a*, three of the resistors terminating on typical node 0, located at a distance $a\,\Delta x$ from the boundary, are calculated as described above. The magnitude of resistor R_1, however, is made a times the magnitude of a corresponding internal resistor (such as R_2) where a is a number less than unity.

It is sometimes expedient to employ a finer net spacing in those portions of the field in which more accurate solutions are required. A simple and direct method for effecting a variation in net spacing is to regard the "coarse" and "fine" regions of the analog



(*a*) (*b*)

FIG. 9.4.4. (*a*) Two-dimensional field with superimposed finite-difference grid. (*b*) Typical node in vicinity of boundary.



(*a*) (*b*)

FIG. 9.4.5. (*a*) Continuous field showing "fine" region in which greater accuracy is required. (*b*) Analog for both portions of field (spacing ratio 1:2).

system as separate and distinct fields, and to treat the interface between these two regions as a streamline boundary. The two networks are then joined at the appropriate nodes. Figure 9.4.5 illustrates how a spacing variation in the ratio of 1:2 may be obtained. Note that the interior resistors of both networks are identical since $\Delta x = \Delta y$ in both sections.

9.4.3. Laplace's Equation in Curvilinear Coordinates. In cylindrical coordinates, Laplace's equation is expanded as

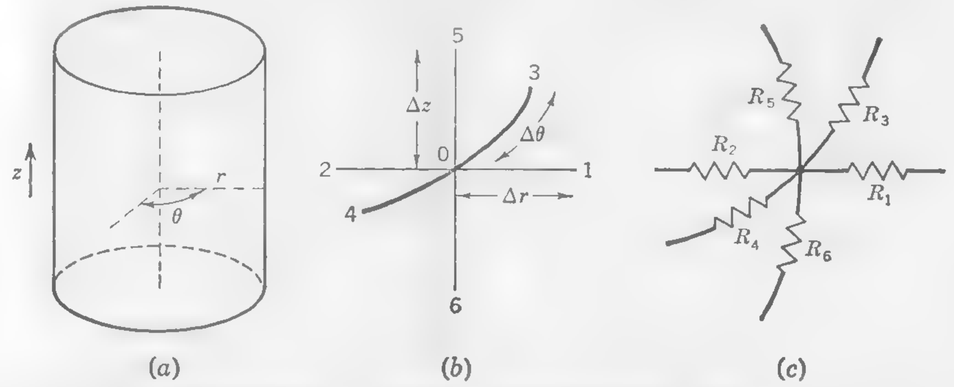
$$\nabla^2\phi = \frac{1}{r}\left[\frac{\partial}{\partial r}\left(r\frac{\partial\phi}{\partial r}\right) + \frac{1}{r}\frac{\partial^2\phi}{\partial\theta^2} + r\frac{\partial^2\phi}{\partial z^2}\right] \tag{9.4.7}$$

The discretized equivalent consists of line segments extending in the r, z, θ directions. A typical node is again the junction of six resistors (Fig. 9.4.6). The finite-difference

approximation to Eq. (7) is

$$\nabla^2\phi = \frac{1}{r_0}\left[\frac{2r_0 + \Delta r}{2\Delta r^2}(\phi_1 - \phi_0) + \frac{2r_0 - \Delta r}{2\Delta r^2}(\phi_2 - \phi_0) + \frac{1}{r_0\,\Delta\phi^2}(\phi_3 - \phi_0) + \frac{1}{r_0\,\Delta\theta_2}(\phi_4 - \phi_0) + \frac{r_0}{\Delta z^2}(\phi_5 - \phi_0) + \frac{r_0}{\Delta z^2}(\phi_6 - \phi_0)\right] \tag{9.4.8}$$

Multiplying through by the scale factor $\Delta r\,\Delta\theta\,\Delta z/R_0$ and comparing with Eq. (5*c*), it



(*a*) (*b*) (*c*)

FIG. 9.4.6. (*a*) Cylindrical field. (*b*) Node of finite-difference grid. (*c*) Typical node of analog network.

is seen that the voltage distribution in a rectangular network, as described by Kirchhoff's current law, will be governed by Eq. (8), provided that

$$R_1 = \frac{R_0\,\Delta r}{\Delta\theta\,\Delta z(r_0 + \Delta r/2)} \qquad R_3 = R_4 = \frac{R_0 r_0\,\Delta\theta}{\Delta r\,\Delta z}$$
$$R_2 = \frac{R_0\,\Delta r}{\Delta\theta\,\Delta z(r_0 - \Delta r/2)} \qquad R_5 = R_6 = \frac{R_0\,\Delta z}{r_0\,\Delta r\,\Delta\theta} \tag{9.4.9}$$

Note that in this case the term r_0 is contained in each expression so that the magnitudes of all resistors are affected by their position within the network. One or two dimensional networks suffice to simulate a problem in cylindrical coordinates, provided that potential symmetry permits one or more of the space coordinates to be neglected. For example, if axial symmetry exists, only the r and z coordinates are significant; so that a two-dimensional resistance network is sufficient. In such fields, resistors located along streamline boundaries may again be given magnitudes twice those of the corresponding interior resistors. More sophisticated techniques for the determination of such boundary resistors are available.[1]

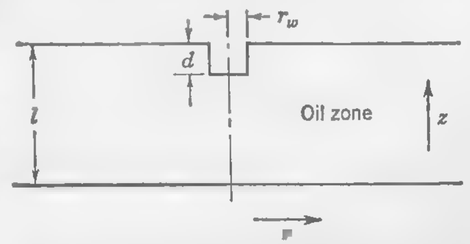


FIG. 9.4.7. Oil well, partially penetrating a horizontal oil zone bounded above and below by impermeable strata.

As an example, consider the flow of an incompressible fluid in the vicinity of an oil well, partially penetrating a horizontal permeable oil stratum bounded above and below by impermeable formations as shown in Fig. 9.4.7. Provided that the well is kept full of oil, it constitutes an equipotential surface of potential ϕ_w. In problems of this type, it is customary to define a drainage radius of r_e at which the flow is essentially horizontal, and a vertical equipotential surface ϕ_e may be taken as a field boundary. The upper and lower surfaces of the oil zone are streamline boundaries. Since the flow field is symmetrical about the well axis, the problem is best formulated in the r and

cylindrical coordinates as

$$\nabla^2\phi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\phi}{\partial r}\right) + \frac{\partial^2\phi}{\partial z^2} = 0 \tag{9.4.10}$$

The potential function ϕ is termed the *velocity potential* and is defined by D'Arcy's law as

$$\phi = \frac{k}{\mu}(P + g\rho z) \tag{9.4.11}$$

where k = hydraulic permeability of porous medium
μ = viscosity
ρ = density
P = pressure

The boundary conditions are expressed as

$$\begin{aligned}
\phi &= \phi_w \quad \text{for} \begin{cases} r = r_w \\ (l-d) < z < 1 \end{cases} \\
\phi &= \phi_e \quad \text{for} \begin{cases} r = r_e \\ 0 < z < l \end{cases} \\
\frac{\partial\phi}{\partial z} &= 0 \quad \text{for} \begin{cases} r_w < r < r_e \\ z = l \quad z = 0 \end{cases}
\end{aligned} \tag{9.4.12}$$

Figure 9.4.8 illustrates a resistance-network analog of such a flow field. The well is assumed to have a radius of one-fourth unit, and a finer net spacing is employed in the upper region of the flow space where larger potential gradients are to be expected. Since a vertical equipotential surface is assumed to exist at the drainage radius, all nodes at the right edge of the resistance network are connected to a common junction. Those nodes on the left side of the network, which correspond to the partially penetrating well, are likewise connected to a common well junction. The penetration of the well can readily be varied or adjusted by suitable selection of the boundary nodes connected to the well junction.

To simulate any given flow condition, it is then only necessary to employ a voltage supply to pass a current proportional to the oil-production rate between the well junction and the drainage-radius junction. The voltages at the network nodes then correspond to the velocity potentials at the corresponding points in the prototype field.

Laplace's equation in spherical coordinates is given by

$$\nabla^2\phi = \frac{1}{r^2}\left[\frac{\partial}{\partial r}\left(r^2\frac{\partial\phi}{\partial r}\right) + \frac{1}{\sin\beta}\frac{\partial}{\partial\beta}\left(\sin\beta\frac{\partial\phi}{\partial\beta}\right) + \frac{1}{\sin^2\beta}\frac{\partial^2\phi}{\partial\alpha^2}\right] \tag{9.4.13}$$

whose finite-difference approximation is given by

$$(\nabla^2\phi)_0 = \frac{1}{r_0^2}\left\{\frac{[r_0 + (\Delta r/2)]^2(\phi_1 - \phi_0)}{\Delta r^2} + \frac{[r_0 - (\Delta r/2)]^2(\phi_2 - \phi_0)}{\Delta r^2} + \frac{\phi_3 - \phi_0}{\Delta\alpha^2\sin^2\beta_0} + \frac{\phi_4 - \phi_0}{\Delta\alpha^2\sin^2\beta_0} + \frac{\sin[\beta_0 + (\Delta\beta/2)](\phi_5 - \phi_0)}{\Delta\beta^2\sin\beta_0} + \frac{\sin[\beta_0 - (\Delta\beta/2)](\phi_6 - \phi_0)}{\Delta\beta^2\sin\beta_0}\right\} \tag{9.4.14}$$

The magnitudes of the analog-network resistors are seen to be

$$\begin{aligned}
R_1 &= \frac{R_0\,\Delta r}{[r_0 + (\Delta r/2)]^2\,\Delta\alpha\,\Delta\beta\sin\beta_0} & R_3 &= \frac{R_0\,\Delta\alpha\sin\beta_0}{\Delta r_s\,\Delta\beta} \\
R_2 &= \frac{R_0\,\Delta r}{[r_0 - (\Delta r/2)]^2\,\Delta\alpha\,\Delta\beta\sin\beta_0} & R_4 &= \frac{R_0\,\Delta\alpha\sin\beta_0}{\Delta r_s\,\Delta\beta} \\
R_5 &= \frac{R_0\,\Delta\beta}{\Delta r\,\Delta\alpha\sin[\beta_0 + (\Delta\beta/2)]} & R_6 &= \frac{R_0\,\Delta\beta}{\Delta r\,\Delta\alpha\sin[\beta_0 - (\Delta\beta/2)]}
\end{aligned} \tag{9.4.15}$$

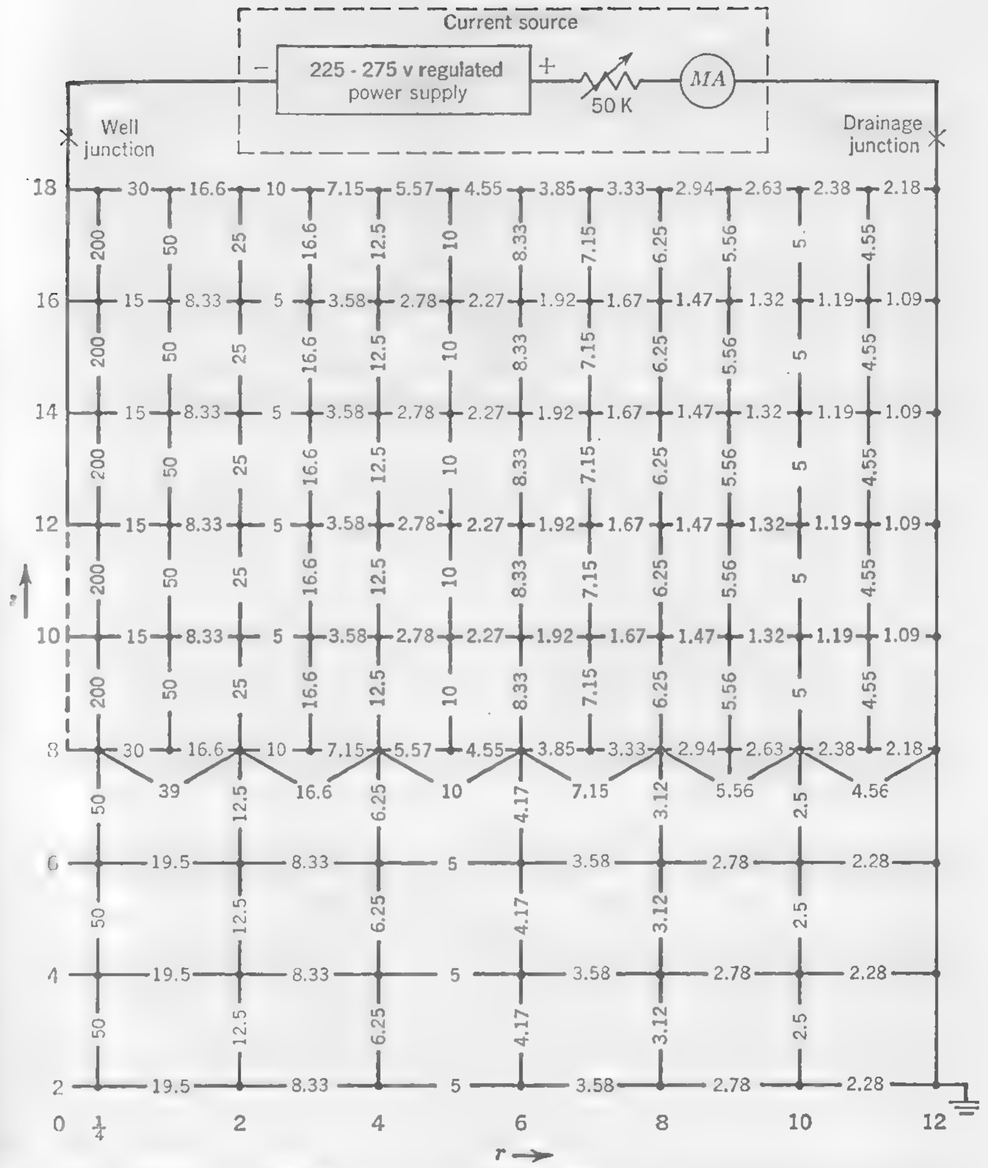


FIG. 9.4.8. Resistance-network analog for field shown in Fig. 9.4.7, using a scale factor $R_0 = 25{,}000$ (numbers represent resistance values in kilohms).

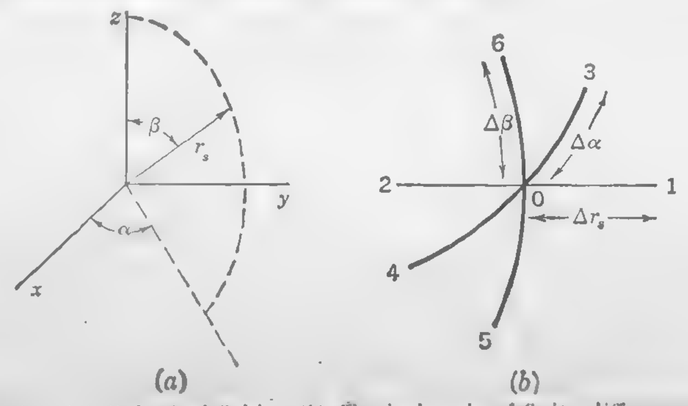


FIG. 9.4.9. (*a*) Spherical field. (*b*) Typical node of finite-difference grid.

Again, all the magnitudes of the resistors are seen to depend upon their distance from the origin. Fields which are symmetrical about a point, so that the α and β coordinates can be neglected, may be represented by a one-dimensional resistance network with typical resistors R_1 and R_2.

9.4.4. Asymmetrical Networks. A few investigators have preferred asymmetric meshes, formed by interconnecting irregularly spaced points in a two-dimensional field, to the more conventional rectangular grid. Following a method described by MacNeal,[6] the node points of the system are first selected. It is permissible in this connection to select arbitrarily a denser spacing of node points in regions of greater interest and to locate an appreciable number of node points on the boundary, which may have an irregular shape. The only restriction upon the number and location of node points is that all the interior angles of the triangles formed by interconnecting the node points be less than 90 deg. A typical grid meeting this requirement is shown in Fig. 9.4.10.

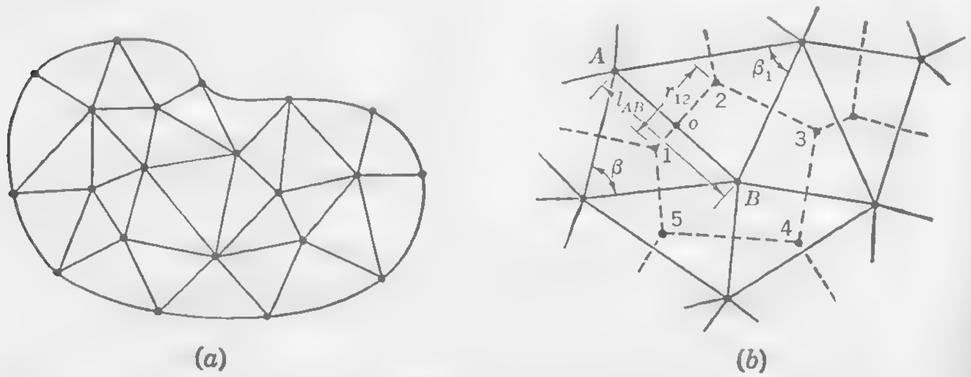


FIG. 9.4.10. (*a*) Asymmetrical finite-difference grid. (*b*) Typical line segment of asymmetrical grid.

To assign magnitudes to the resistors representing each grid line, the perpendicular bisectors are constructed for each line segment. Each node point is thereby surrounded by a polygon formed by the bisectors of the grid lines terminating on that point. For a typical grid line connecting points A and B, having a length l_{AB}, and a perpendicular bisector (joining points 1 and 2) of length r_{12}, the value of the resistor connecting A and B is given by

$$R_{AB} = \frac{R_0 l_{AB}}{r_{12}} \tag{9.4.16}$$

where R_0 is again a scale factor.

9.4.5. Analog for Equations of the Poisson Type. Field problems occurring in many areas of engineering and applied physics are governed by an equation of the type

$$\nabla^2\phi = f(x,y,z,\phi) \tag{9.4.17}$$

In cartesian coordinates and three dimensions the finite-difference approximation of this equation is

$$\frac{\phi_1 - \phi_0}{\Delta x^2} + \frac{\phi_2 - \phi_0}{\Delta x^2} + \frac{\phi_3 - \phi_0}{\Delta y^2} + \frac{\phi_4 - \phi_0}{\Delta y^2} + \frac{\phi_5 - \phi_0}{\Delta z^2} + \frac{\phi_6 - \phi_0}{\Delta z^2} = f(x,y,z,\phi_0)$$

or multiplying by $(\Delta x\,\Delta y\,\Delta z/R_0)$

$$\frac{\Delta y\,\Delta z}{\Delta x\,R_0}(\phi_1 - \phi_0) + \frac{\Delta y\,\Delta z}{\Delta x\,R_0}(\phi_2 - \phi_0) + \frac{\Delta x\,\Delta z}{\Delta y\,R_0}(\phi_3 - \phi_0) + \frac{\Delta x\,\Delta z}{\Delta y\,R_0}(\phi_4 - \phi_0)$$
$$+ \frac{\Delta x\,\Delta y}{\Delta z\,R_0}(\phi_5 - \phi_0) + \frac{\Delta x\,\Delta y}{\Delta z\,R_0}(\phi_6 - \phi_0) = \frac{\Delta x\,\Delta y\,\Delta z}{R_0} f(x,y,z,\phi_0) \tag{9.4.18}$$

Referring to the electric analog equation, Eq. (5*c*), it is seen that each term on the left side of these equations has the dimension of current. It appears, therefore, that the terms on the right side of Eq. (19) must likewise be represented by a current, so that the analog equation becomes

$$\frac{V_1 - V_0}{R_1} + \frac{V_2 - V_0}{R_2} + \frac{V_3 - V_0}{R_3} + \frac{V_4 - V_0}{R_4} + \frac{V_5 - V_0}{R_5} + \frac{V_6 - V_0}{R_6} = i(x,y,z,V_0) \tag{9.4.19}$$

To simulate problems governed by Eq. (17), the Laplacian is simulated by a rectangular resistance network exactly as described above. The term on the right side is then taken into account by introducing a current which may be a function of position and/or voltage at each node of the resistance network, as shown in Fig. 9.4.11*a*. To avoid the use of current sources, it is usually preferred to connect each node to a fixed voltage source E_f through a variable resistor R_f, as shown in Fig. 9.4.11*b*. These variable feed-in resistors are then adjusted until the correct current flows into each node point. Since the current into each node affects, to some extent, the current into

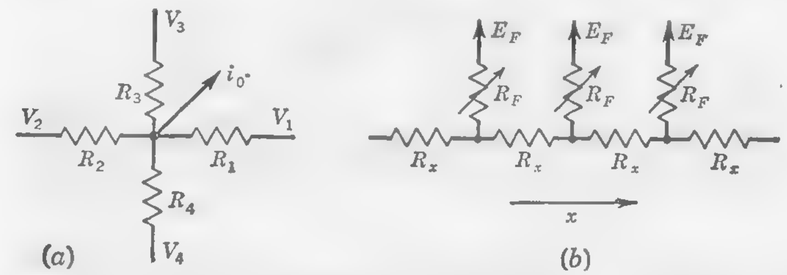


FIG. 9.4.11. (*a*) Typical node of resistance-network analog for Poisson-type field. (*b*) Resistance-network analog for two-dimensional field using feed-in voltage sources.

every other node, the adjustment of these feed-in resistors must proceed iteratively and may become very time-consuming. Analog-computor units[1] as well as various other automatic units have been designed to perform this iteration rapidly and accurately.

9.4.6. Analog for the Diffusion Equation. The analog determination of the transient potentials in fields governed by the diffusion equation

$$\nabla^2\phi = K\frac{\partial\phi}{\partial t} \tag{9.4.20}$$

is usually accomplished using networks containing resistors as well as capacitors. It is possible to determine these potentials using network analogs containing resistors only, by discretizing time as well as the space coordinates. That is, the time variable is broken up into discrete intervals Δt in length. Equation (20) in one cartesian space dimension is then approximated by

$$\frac{\phi_1 - \phi_0}{\Delta x^2} + \frac{\phi_2 - \phi_0}{\Delta x^2} = K\frac{\phi_0 - \phi_{0-\Delta t}}{\Delta t} \tag{9.4.21}$$

The symbol $\phi_{0-\Delta t}$ represents the potential existing at node zero at a time Δt before the time at which the calculation is currently being made. This equation is seen to be similar in form to Eq. (17) and has a resistance-network analog whose governing equation is

$$\frac{V_1 - V_0}{R_x} + \frac{V_2 - V_0}{R_x} = \frac{V_0 - V_{0-\Delta t}}{R_t} \tag{9.4.22}$$

A typical node of the analog network is shown in Fig. 9.4.12*a*, while a complete one-dimensional system is illustrated in Fig. 9.4.12*b*. This network permits the prediction of the potentials at all nodes at a time $t = t_0 + \Delta t$ if the potentials at time $t = t_0$ are known. To solve a typical problem, the resistors R_t are made to correspond to a convenient Δt, and the given initial voltages at time $t = 0$ are applied at their lower terminals. The potentials at time $t + \Delta t$ may then be obtained by measuring the voltages at the network nodes. The feed-in voltages to all feed-in resistors are then readjusted to correspond to the network voltages for time $t = t_0 + \Delta t$. The voltages now appearing at the network nodes correspond to the potentials at time $t = t_0 + 2\Delta t$. The feed-in voltages are then again adjusted, and the process is continued until the potentials corresponding to the desired number of time intervals have been obtained. The finite-difference approximation equation (21) has the advantage over other numerical schemes in that there exists no danger of computational instability, regardless of how large the time spacing Δt is made.

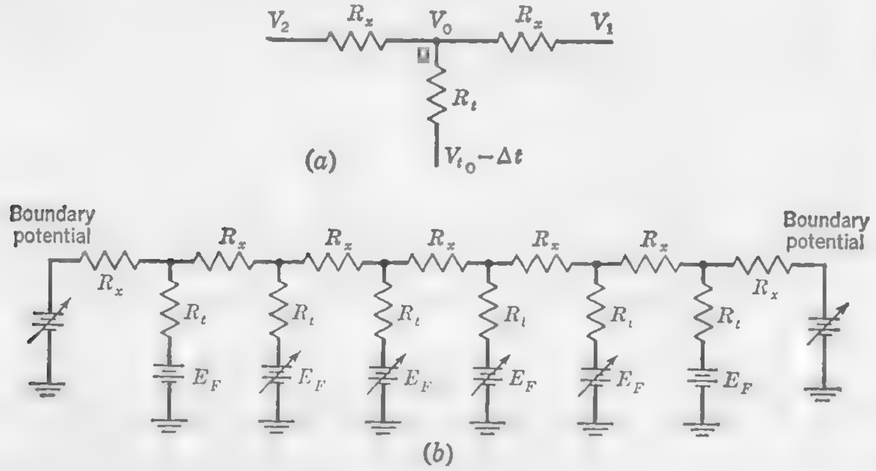


FIG. 9.4.12. (*a*) Typical node of resistance-network analog for one-dimensional diffusion equation. (*b*) One-dimensional analog system using feed-in voltages.

This method can be employed for linear and nonlinear problems in one, two, or three space dimensions and is capable of complete automation. In such an automatic system, the adjustments of the feed-in voltages and all read-outs are accomplished by servo units. It is then only necessary to apply the specified initial conditions and to specify the time increment, and the computer will print out the correct field potentials at discrete time and space intervals.

9.4.7. Analog for the Wave Equation. Fields governed by the wave equation

$$\nabla^2\phi = K\frac{\partial^2\phi}{\partial t^2} \tag{9.4.23}$$

are frequently simulated by means of inductance-capacitance networks. It is possible, however, to adapt the method described above for simulating fields governed by the diffusion equation to fields governed by the wave equation as well. Both the time and the space variables are again discretized and an implicit finite-difference scheme is employed to avoid computational instability. In problems governed by Eq. (23), two initial conditions, generally the potential as well as the first derivative with respect to time of the potential everywhere within the field, are given as part of the problem specification. In effect, therefore, one has available the potential function at time $t = t_0$ and at time $t = t_0 + \Delta t$; potentials for time $t = t_0 + 2\Delta t$ and subsequent time increments are to be determined. Two voltages must therefore be applied as feed-in sources to each network node. The solution over any given period of time is then obtained in an iterative manner by measuring the node potentials corresponding to each time increment and by employing these potentials as "initial conditions" for the next time increment.

If the boundaries of a field governed by the wave equation are kept at fixed potentials, and energy is present within the system at time $t = 0$, the node potentials undergo simple harmonic variations, and a condition of periodic steady state exists. The potential at each node then has the form

$$\phi = A\sin\omega t$$

so that

$$\frac{\partial^2\phi}{\partial t^2} = -A\omega^2\sin\omega t = -\omega^2\phi \tag{9.4.24}$$

and Eq. (22) becomes under these conditions

$$\nabla^2\phi = -K\omega^2\phi \tag{9.4.25}$$

Equation (25) has the same general form as Eq. (17); so that this problem can be treated just like field problems governed by equations of the Poisson type, discussed above. The analog model therefore contains a rectangular resistance network with a current $i_0 = K\omega^2\phi_0$ introduced at each node. This method is particularly suitable for the determination of the normal-mode values.

9.4.8. Analog for the Biharmonic Equation. Plane-stress problems governed by the biharmonic equation in two dimensions

$$\nabla^4\phi = \frac{\partial^4\phi}{\partial x^4} + \frac{2\partial^4\phi}{\partial x^2\partial y^2} + \frac{\partial^4\phi}{\partial y^4} = 0 \tag{9.4.26}$$

are generally treated by the direct analogy described in Sec. 9.1. Resistance-network methods can be applied to this equation, however, by fashioning two separate rectangular resistance networks and by interconnecting them at each node as shown in Fig. 9.4.13.

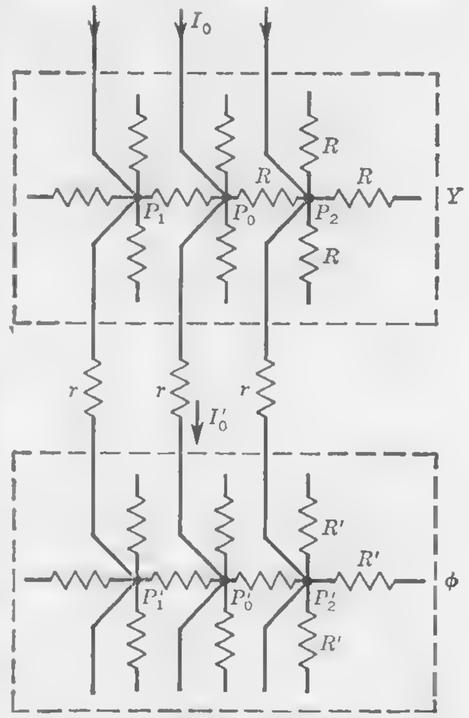


FIG. 9.4.13. Basic arrangement of cascaded resistance networks for solution of biharmonic equation.

The potentials at the nodes of the upper network represent a stress function Y defined as

$$Y = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} \tag{9.4.27}$$

whose value must be given along the boundaries of the field under study. The node voltages of the lower resistance network represent the desired stress function ϕ. The current I_0' through the resistor r connecting the corresponding nodes of the two networks is then given by

$$I_0' = \frac{Y_0 - \phi_0}{r} \tag{9.4.28}$$

If the connecting resistors r are much larger than the resistors making up the rectangular resistance networks, then for any particular pair of nodes

$$Y_0 \gg \phi_0 \qquad \text{and} \qquad I_0' = \frac{Y_0}{r} \tag{9.4.29}$$

The equation simulated by the lower network is therefore

$$\nabla^2\phi = k\frac{Y_0}{r} \tag{9.4.30}$$

where k is a constant of proportionality; but since the voltage distribution Y satisfies Laplace's equation,

$$\nabla^2(\nabla^2\phi) = \nabla^4\phi = 0 \tag{9.4.31}$$

The potentials of the lower network therefore represent the solution to the problem. Problems in which $\nabla^4\phi = k$ are solved with a similar resistance-network arrangement, by introducing a current proportional to k at each node of the upper network.

CONDUCTIVE-SOLID ANALOGIES

9.4.9. Principle of the Analogy. The electrical potential and current distribution in an electrical conductor are governed by Laplace's equation, Eq. (1). Because they are convenient to construct and to use, electrically conductive sheets and solids are frequently used as simulators. The application of such analogs to Laplacian field problems then takes the following steps:

1. A conductive sheet or solid having the same geometrical shape as the two-dimensional field under study is devised.
2. The boundary conditions of the original field are simulated in the electrical-analog system by means of suitable voltage and/or current sources.
3. By means of suitable sensing equipment the voltage distribution within the conductive medium is measured and recorded. The voltages obtained in this manner can then be interpreted directly as the potential distribution in the original system under study.

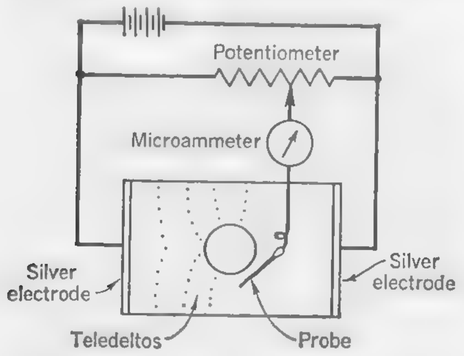


FIG. 9.4.14. Simple conductive-sheet analog system.

A typical conductive-sheet analog system is shown in Fig. 9.4.14. Numerous engineering problems can be formulated adequately in two cartesian dimensions or in polar coordinates. For such problems, a two-dimensional electrical analog—a conductive sheet—is adequate. Problems which can be formulated only in three cartesian dimensions require three-dimensional conductive analogs. A chief difficulty of the latter type of system is that it is generally impossible to measure the potential distribution at interior points of the analog. The use of conductive-solid analogs is therefore restricted to problems in which the potential distribution at the exterior surfaces of the physical system is required. The conductive-liquid analogs described in the following section do not suffer from this shortcoming.

9.4.10. Resistance Paper—Teledeltos. By far the most widely used conductive-solid analog is a type of electrosensitive recording paper developed by the Western Union Telegraph Co. This material is now widely used as a recording medium for telegrams as well as for graphic-chart-recording instruments. Known by the trade name Teledeltos, this paper is formed by adding carbon black, a conductive material, to paper pulp in the pulp-beating stage of the paper-manufacturing process. The conductive paper is then coated on one side with a lacquer which acts as an electrical insulator and on the other side with a thin layer of aluminum paint. Two basic types of Teledeltos are available: type L, a low-resistivity paper ranging between 1,500 and 4,000 ohms per square, and type H, the high-resistivity paper with a resistivity of approximately 20,000 ohms per square. No precise control of the sheet resistivity is exercised during manufacture; so that there may be a wide variation in resistivity from roll to roll. Furthermore, uniformity tests have revealed that a variation in resistivity

of approximately ±10 per cent may be expected in any given sheet. The manufacturing process also introduces a definite anisotropicity, the resistivity measured along the long dimension of a roll generally being 10 per cent lower than the resistivity across the width of the roll.

Teledeltos paper has a temperature coefficient of resistance of approximately −0.2 per cent per °C so that it is relatively insensitive to changes in temperature. Relatively large changes of resistance, sometimes observed when the paper is heated (possibly by the electric currents introduced during analog simulation), are generally due to changes in the moisture content of the paper.

Notwithstanding its shortcomings as far as homogeneity and uniformity are concerned, Teledeltos paper has found great favor as a conductive-sheet analog for the solution of field problems in which accuracies to within approximately 10 per cent are sufficient. This popularity may largely be attributed to the convenience and simplicity inherent in the use of resistance paper. Teledeltos is readily available in rolls up to 35 in. in width at a relatively low price. It is no harder to cut with scissors than ordinary paper and may readily be cemented to glass or plastic backing plates. Electrode boundaries are readily applied by means of conductive paint, such as Du Pont No. 4817 silver paint. Local changes in the resistivity of the conductive sheet as dictated by the problem to be solved may be attained by perforating the paper or by applying several layers of a graphite solution.

9.4.11. Conductive Rubbers. Rubber sheets having an appreciable electrical conductivity are now manufactured for a number of industrial applications. Electrical conductivity is imparted to the material by adding specially processed carbon black to the rubber stock during the manufacturing process. From the point of view of homogeneity and uniformity, high-quality conductive rubbers compare favorably with Teledeltos paper. The resistivity of typical sheets of approximately 0.01 in. thickness varies between 400 and 2,000 ohms per square. A major advantage of conductive sheets of this type is that several layers can readily be pressed against each other and in many cases adhere under pressure. This permits the simulation of fields of variable resistivity and the study of the potential distribution at the external surfaces of three-dimensional fields. Most conductive rubbers display an anisotropicity of approximately the same order of magnitude as Teledeltos paper.

9.4.12. Miscellaneous Conductive Models. A wide variety of materials have been employed from time to time as conductive-solid analog simulators.[1] Metallic sheets were used as early as 1845 for the simulation of Laplacian fields. The chief difficulty with this type of model is the obtaining of uniform resistivity. Since the local resistivity of metallic sheets is strongly influenced by changes in thickness of the sheets, it is difficult to reduce errors below 10 per cent. The very low resistivity of metallic sheets presents a further difficulty. Electrodes comprising the equipotential boundaries must have a resistivity which is negligible compared with that of the conductive sheet. This requirement is very hard to satisfy when metallic sheets are employed.

Ordinary writing paper has occasionally been used as a conductive-sheet analog. Papers made of cellulose or rags having a fiber structure are severely anisotropic, while transparent drawing paper does not suffer from this shortcoming. The conductivity of the paper appears to be due principally to absorbed water which with dissolved salts forms a uniform electrolyte. As a result the conductivity of untreated paper is strongly affected by the atmospheric humidity. Because of the extremely high resistivity of such sheets, equipotential boundaries may be applied simply by drawing pencil lines on paper. The sensing device must have an extremely high input impedance, and electrometer tubes are ordinarily employed.

A number of investigators have preferred to develop suitable conductive-sheet models in the laboratory. These generally consist of a nonconducting base coated with a layer of conductive material.[1]

9.4.13. Analog for Poisson's Equation. In a number of areas of engineering, particularly in torsional-stress analysis, there arise equations of the type

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = k \tag{9.4.32}$$

The simulation of fields governed by Eq. (32) would ordinarily require distributed current inputs over the entire area of the conductive sheet. It is possible to solve such problems, using field models suitable only for the simulation of Laplacian fields, by effecting a change of variables such that the potential function ϕ is replaced by ψ where

$$\phi = \psi + \frac{k}{4}(x^2 + y^2) \tag{9.4.33}$$

Equation (32) then becomes

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{9.4.34}$$

The boundary conditions of the original problem must be similarly transformed. For example, if the boundary potential is specified to have a constant value ϕ_B along some boundary, the corresponding boundary condition in the transformed problem becomes

$$\psi_B = \phi_B - \frac{k}{4}(x^2 + y^2) \tag{9.4.35}$$

where x and y represent the coordinates of the boundary. A number of techniques are available for the realization of continuously varying boundary potentials in conductive sheets.[1]

CONDUCTIVE LIQUIDS—THE ELECTROLYTIC TANK

9.4.14. Principle of Operation. The basic principles underlying the operation of conductive-liquid analog systems are similar to those governing the conductive-sheet and solid analogs described above. The conductive material in this case is water containing small amounts of dissolved and ionized salts. Liquids of this type are termed electrolytes. If, by means of suitable precautions, capacitive reactance components in the system are effectively eliminated, the liquid has the properties of a pure conductor, and it may be employed to simulate any field governed by Laplace's equation.

The utilization of electrolytic field models was first suggested in 1875, and until the development of Teledeltos paper and the popularization of resistance-network analogs after World War II, the electrolytic tank was by far the most popular and widely used electrical-analog system for the simulation of potential fields. The solution of field problems governed by Eq. (1) proceeds as follows:

1. A large container, usually open at the top, is filled with a suitable electrolyte.
2. A scale model of the boundary configuration of the field under study, or a conformal transformation thereof, is immersed in the tank. Boundaries which are equipotential surfaces are made of metal while an insulating material is employed for streamline boundaries.
3. A-C voltage sources of appropriate magnitudes are applied to all equipotential boundaries.
4. By means of suitable sensing equipment, the voltage distribution along the surface of the liquid is measured and recorded. If Laplace's equation in three dimensions is required to describe the field, the sensing probe may be extended into the liquid and a three-dimensional record of the potential distribution within the tank obtained. Lines of constant voltage within the conductive analog then correspond directly to the equipotential lines of the original system.

9.4.15. Design of Electrolytic Tanks. A typical conductive-liquid analog system is illustrated in Fig. 9.4.15. This type of simulator is seen to include six major units: the conductive liquid, the boundary and electrode configurations, the electrical excitation unit, the probe, the sensing unit, and the recording device.

Weak aqueous solutions of common salts (for instance, 3 parts copper sulfate per 1,000 parts of distilled water) are employed almost exclusively as conductive liquid. In order to constitute a satisfactory analog for Laplacian fields, the conductive liquid should have negligible reactive properties and have a resistivity which is uniform and

linear throughout. Polarization effects in the immediate vicinity of the electrodes account for the principal source of error introduced by the electrolyte. Even when the exciting frequencies are in excess of several hundred cycles per second, nonlinear resistance and capacitive effects appear at the electrode surfaces. For maximum accuracy it is necessary to choose the combination of liquid and electrode material very carefully. While platinum or platinized electrodes in combination with copper sulfate solutions appear to be optimum from this point of view, copper, brass, and graphite electrodes have been found adequate for many applications. The meniscus of the surface of the electrolyte may cause distortions of the potential field in the immediate vicinity of the electrodes as well as at the probe. Chemical additives to reduce surface tension are frequently included in the electrolyte to minimize errors from this source.

The construction of the electrolytic tank is governed to a large extent by the geometry and the nature of the field problem under investigation. Most generally the main container is rectangular and fashioned of wood. The inside of the tank is carefully coated with waterproofing lacquer and waxed to prevent leakage and to avoid warping and swelling. Detailed shaping of the tank walls, where required, is generally accomplished with wax.

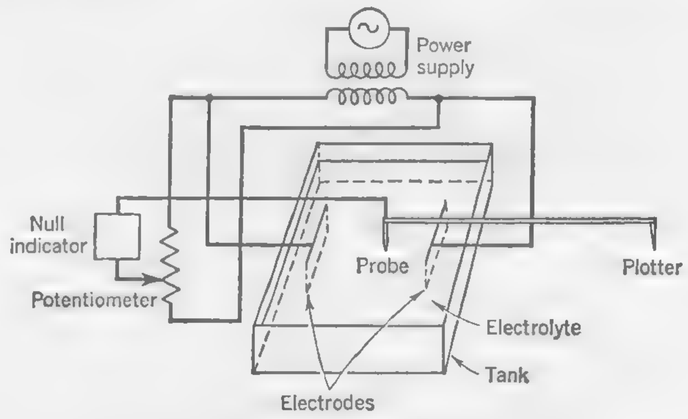


FIG. 9.4.15. Electrolytic-tank analog system.

For two-dimensional problems in cartesian coordinates, the bottom of the tank is flat, so that the depth of the electrolyte is constant. If the potential field is symmetric about a line, so that the cylindrical coordinates r and z are appropriate, a sloping bottom may be employed. The conductive liquid then simulates a typical wedge cut out of the original field. A flat-bottom shallow electrolytic tank may be employed for the same purpose by tilting it in such a manner that the liquid-air interface intersects the bottom of the tank. In either case, errors may arise near the apex of the wedge because of the meniscus of the electrolyte-bottom interface. These can be minimized by coating those portions of the tank bottom which are not to be covered by the electrolyte with a water-repellent substance.

The electrical excitation of the electrolytic-tank system is nearly always in the form of a-c voltages of constant frequency. Frequencies in the vicinity of 1,500 cps appear to be most satisfactory. The magnitude of the sinusoidal excitation must be large enough to ensure a satisfactory signal-to-noise ratio at the probes, but the current density within the electrolyte must be restricted to obviate undesirable electrochemical effects and to prevent convection currents in the electrolyte resulting from the heating of the liquid. Square-wave excitations have been used on occasion.

The electrical field within the tank is sensed by means of a suitable probe or an array of probes. Most frequently the equipotential lines are to be mapped. In that case a single-probe tip is immersed into the electrolyte and its voltage with respect to one of the electrodes or with respect to some arbitrary ground potential is recorded, usually by potentiometric methods. The direct determination of the field strength

and the plotting of streamlines require a more elaborate setup. Generally a four-probe array is employed for this purpose. Numerous and elaborate electronic units have been devised for the sensing and recording of the field potentials. These include automatic field plotters to produce complete families of equipotential lines rapidly and accurately.[1]

Provided that all the sources of error are carefully minimized, accuracies within 0.1 per cent may be obtained with electrolytic-tank analogs. Such accuracies are attainable only with extreme care and meticulous adjustment, however. Most conductive-liquid analog systems of the type used in industrial laboratories have accuracies no better than within 2 per cent, which is adequate for many engineering purposes.

STRETCHED-MEMBRANE ANALOGIES

9.4.16. Membrane Analogies. A very important class of analog simulators for fields governed by Laplace's and Poisson's equations employ transverse deflections of elastic members as the dependent variable. It can be shown that, when a thin elastic sheet, located in the horizontal x, y plane, is placed under uniform tension and attached to a flat frame, the sheet adjusts itself to any vertical deformation in the frame by taking a shape governed by

$$\frac{\partial^2 z}{\partial x^2}\left[1+\left(\frac{\partial z}{\partial x}\right)^2\right]+\frac{\partial^2 z}{\partial y^2}\left[1+\left(\frac{\partial z}{\partial y}\right)^2\right]-2\frac{\partial^2 z}{\partial x\,\partial y}\frac{\partial z}{\partial x}\frac{\partial z}{\partial y}=0 \qquad (9.4.36)$$

where z is the position of the membrane in vertical direction. If the vertical displacements of the sheet are kept small, so that the slope of the sheet is nearly horizontal everywhere, then

$$\frac{\partial z}{\partial x}\approx 0 \qquad \frac{\partial z}{\partial y}\approx 0 \qquad (9.4.37)$$

and Eq. (37) becomes

$$\frac{\partial^2 z}{\partial x^2}+\frac{\partial^2 z}{\partial y^2}=0 \qquad (9.4.38)$$

The vertical deflection of the membrane is therefore governed by Laplace's equation in two dimensions. The analogy is exact only if the following conditions are satisfied:

1. The tension (force in the horizontal direction) is uniform everywhere throughout the sheet.
2. The tensile force everywhere within the sheet is large compared with the vertical forces due to the weight of the sheet. This implies that the sheet must be very light and thin.
3. The slope of the sheet is small at all points. If the slope is less than 6 deg, the error introduced by this will generally be less than ± 1 per cent.

If the stretched membrane discussed above is excited by a force acting in the z or transverse direction, and if this force is uniformly distributed throughout the sheet, the deflection of the membrane is governed by

$$\frac{\partial^2 z}{\partial x^2}+\frac{\partial^2 z}{\partial y^2}=\frac{q}{T} \qquad (9.4.39)$$

where q is the transverse force per unit area acting in the positive z direction, and T is the uniform tension per unit length of the boundary of the sheet. The transverse load therefore acts as a distributed internal excitation, and the system may be used to simulate two-dimensional fields governed by Poisson's equation.

Two types of stretched-membrane analog systems have found wide application. *Rubber sheets* are used primarily in electrostatic and electron optics-design problems. In these areas they offer the advantage that electron trajectories of interest can readily be detected and mapped by means of small rolling balls. *Soap-film* analogs permit the convenient application of uniformly distributed transverse loads and are therefore very useful for the study of elastic-torsion problems governed by Poisson's equation. If sufficient care is taken in the machine-shop construction of the frames and measuring devices, and if refined experimental techniques are employed, stretched-membrane analogs are capable of accuracies within 1 or 2 per cent and compare not unfavorably with the electric analog simulators.[1,2] A commonly used formula includes 30 cc of glycerin and 2 g of potassium oleate in 1 liter of water; useful film life may be as long as 20 hr, if the solution is kept free from impurities.

References

1. Karplus, W. J.: *Analog Simulation: Solution of Field Problems*, McGraw-Hill Book Company, Inc., New York, 1958.
2. Karplus, W. J., and W. W. Soroka: *Analog Methods: Computation and Simulation*, rev. ed., McGraw-Hill Book Company, Inc., New York, 1959.
3. Liebmann, G.: Electrical Analogues, *Brit. J. Appl. Phys.*, **IV**:193–200 (1953).
4. Liebmann, G.: A New Electrical Analog Method for the Solution of Transient Heat Conductor Problems, *Trans. ASME*, **78**:655–665 (1956).
5. Liebmann, G.: Solution of Plane Stress Problems by an Electrical Analogue Method, *Brit. J. Appl. Phys.*, **VI**:145–157 (1955).
6. MacNeal, R. H.: Asymmetrical Finite Difference Network, *Quart. Appl. Math.*, **11**: 295–310 (1953).

PART 5: HYDRODYNAMIC ANALOGIES

By WALTER W. SOROKA

ANALOGIES FOR TWO-DIMENSIONAL POISSON AND LAPLACE FIELDS

9.5.1. Introduction. The velocity potential in ideal (inviscid) two-dimensional fluid flow and the pressure distribution in viscous flow, both with distributed sources, satisfy the two-dimensional Poisson equation. In the absence of sources the fields satisfy the two-dimensional Laplace equation. Such fluid-flow fields may serve as analogs for other field problems. Examples of such fields follow.

9.5.2. Hydrodynamic Analogies for Prismatic Bars in Torsion. A representation due to Kelvin makes use of a flat tank of unit depth, with cross section shaped like that of the bar to be twisted.[1] The tank is waxed inside, filled with darkened fluid (e.g., water with india ink or potassium permanganate added), and the water surface sprinkled with aluminum particles. The tank is abruptly rotated about the "center of shear," and a photograph of the water surface is taken by an overhead camera rotating with the tank. The streaks in the photograph show the directions of the shear stresses, their lengths being proportional to shear-stress magnitudes. For quantitative purposes, the observed streak lengths may be compared with those applicable to a circular bar twisted through the same angle, for which the shear stresses are known.

9.5.3. Sandbed Mappers. Another representation[2] makes use of a high-flow-resistance horizontal bed of fine nickel or copper shot, through which a real (viscous) fluid slowly percolates vertically to give the required uniform distribution of sources over the surface of the bed. This so-called "sandbed" is contained in a metal box having a cross section shaped like that of the bar to be twisted. It is supported by a fine-mesh screen over a fluid-supply chamber. The horizontal viscous-flow region satisfying the Poisson equation is contained in the thin space between the surface of the bed and a glass plate placed about $\frac{1}{16}$ in. above it. Dye crystals in the flow field produce visible flow lines, which are photographed through the glass plate. Equal-pressure contours are orthogonal to the flow lines and may be drawn on the photograph by the method of curvilinear squares.[3] Tangents to the equal-pressure contours give the directions of the shear stresses. The relative spacings of the contours (propor-

tional to pressure gradient) give the relative values of the shear stresses, the actual values for a given problem being determined either from a measurement of the flow quantity or from a determination of pressure gradients on a circular sandbed tested simultaneously (Fig. 9.5.1).

The sandbed is immersed in a pan of the fluid (usually water to which a small amount of Methocel had been added to increase viscosity). The flow across the boundaries of the box and into or out of the sandbed (preferably *into* the sandbed to avoid dislodging the fine particles of shot at the surface) is caused by lowering or raising an auxiliary container of the fluid connected through rubber tubing to the plenum chamber underneath the sandbed. Nickel shot, because of its roundness, smoothness, and uniformity, is preferred for sandbeds. Copper shot must be sifted to obtain size uniformity.

In another form of the analogy, the sandbed is replaced by a solid plate, which is then moved slowly to either increase or decrease the separation between it and the glass plate above it.[4] This relative movement produces flows over the model surface equivalent to those caused by a sandbed.

It can be shown[5] that the equations for the laminar flow of a viscous fluid in the annular space between two cylindrical tubes eccentrically positioned are also those for

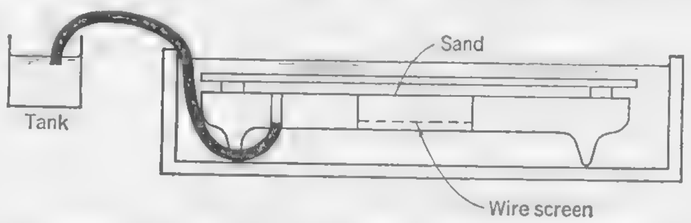


FIG. 9.5.1. Fluid mapper with sandbed to simulate distributed excitation. (*From W. J. Karplus, Analog Simulation, McGraw-Hill Book Company, Inc., New York, 1958.*)

the torsion of a hollow prismatic bar having the same annular cross section. By means of this analogy the dual relationships of flow quantity in the hydrodynamic case and torsional stiffness in the elastic-solid case have been determined as functions of the eccentricity and the ratio of the inner and outer radii.[5]

9.5.4. Hydrodynamic Analogy for Shear Stress in Bending. Neglecting Poisson-ratio effects (usually small), the shear-stress distribution over the cross section of a beam in bending is represented by the velocity distribution in an ideal (inviscid) liquid contained in a two-dimensional tank having the form of the beam section.[6] The flow is developed by distributed positive sources on one side and negative sources on the other side of the neutral axis of the beam section. The strengths of the sources are proportional to their distances from the axis; so that the flow into the negative sources exactly equals the flow out of the positive sources.

9.5.5. Hydrodynamic Analogies for Other Field Problems. Laplacian and Poissonian fields of various types may be developed by using a hard-plaster ("dental stone") lower slab in the *fluid mapper*.[2] This slab may then be provided with one or more simple wells of any shape or size as plus or minus sources, or with a sandbed occupying a fraction of lower surface. Different rates of flow may be developed in the different wells. To represent variable permeability (as in magnetic fields or in oil fields), variable dielectric constant (as in electric fields), variable thermal conductivity (as in heat-conduction studies), or variable diffusivity (as in diffusion studies), etc., the flat-cast lower slab may be machined to give the necessary variation in flow-space thickness. Impervious obstacles of metal, plaster, or rubber may be introduced in the flow field. In magnetic-field studies permeability changes as high as 1,000:1 have been represented in fluid mappers.[7]

HYDRODYNAMIC REPRESENTATION OF CREEP AND RELAXATION IN COMPLEX MATERIALS

9.5.6. Representation of Creep and Relaxation. Conventional viscoelastic representation of creep and relaxation in materials is limited to exponential relations. Real

relations, such as those in prestressed concrete and other substances, may be simulated by the time-dependent forced motion of a piston in a fluid-filled cylinder forcing liquid into a standpipe whose cross-sectional area is a predetermined function of position.[8] The force on the piston is caused to vary with position, as in the case of a spring. Results from such hydraulic models have been found to agree with actual creep and relaxation tests on high-tensile steel wires.[8]

A HYDRAULIC ANALOGY FOR TRANSONIC AERODYNAMICS

9.5.7. A Hydraulic Analogy for Transonic Flow. A compressible-gas flow field with *weak* shock waves (e.g., in the transonic regime) may be simulated by a thin (optimally just under 0.25-in.) layer of water through which a model is towed.[9] The analogy applies to a hypothetical gas for which the specific-heat ratio is 2. However, the analogy is used to develop or to verify analytical procedures which may be used for real gases. In this analogy the squares of the relative depths of the thin water layer distributed by the passing of a towed model represent relative pressures in the "hydraulic-gas" flow at corresponding points in the two-dimensional field about a similar obstacle. Water depths are measured with resistance probes of 0.004-in.-diameter piano wire.[10]

References

1. Den Hartog, J. P., and J. G. McGivern: On the Hydrodynamic Analogy of Torsion, *J. Appl. Mechanics*, **2**:46–48 (1935).
2. Moore, A. D.: The Further Development of Fluid Mappers, *Trans. AIEE*, **69**(II): 1615–1624 (1950).
3. Moore, A. D.: Mapping Techniques Applied to Fluid Mapper Patterns, *Elec. Eng.*, **71**(5):446 (1952).
4. Pestel, E.: Eine neue hydrodynamische Analogie zur Torsion prismatischer Stäbe, *Ing.-Arch.*, **23**(3):172–178 (1955).
5. Shevelov, Ya. V.: Poiseuille's Flow in Nonsymmetrical Radial Clearance; Analogy with the Torsion of Beams (in Russian), *Doklady Akad. Nauk SSSR*, **91**(1):35–38 (1953).
6. Biot, M. A.: A Hydrodynamic Analogy for Shearing Stress Distribution in Bending, *J. Appl. Phys.*, **9**:39–43 (1938).
7. Hele-Shaw, H. S.: Lines of Induction in a Magnetic Field, *Electrician*, **52**:213–215, 307–310, 350–352 (1904).
8. Olszak, W., and J. Litwiniszyn: Non-linear Liquid Flow as a Rheological Model (in English), *Polska Akad. Nauk, Classe* 4, **2**(2):69–73 (1954).
9. Bryant, R. A. A.: The Hydraulic Analogy as a Distorted Dissimilar Model, Readers' Forum, *J. Aeronaut. Sci.*, **23**(3):282–283 (1956).
10. Laitone, E. V.: Developments in Gas Dynamics by the Hydraulic Analogy (Mémoires sur la mécanique des fluides), *Publs. sci. et tech. ministère air* (*France*), 1954, pp. 203–217.

See also Karplus, W. J., *Analog Simulation*, McGraw-Hill, Book Company, Inc., New York, 1958, and Karplus, W. J., and W. W. Soroka, *Analog Methods in Computation and Simulation*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1959.

DIGITAL COMPUTERS

Section 10

COMPONENTS

By

ISAAC L. AUERBACH, *President and Technical Director, Auerbach Electronics Corporation, Philadelphia, Pa.*

J. JAMES EBERS,* *Formerly with Bell Telephone Laboratories, Inc., Allentown Laboratory.*

M. L. EMBREE, *Member of Technical Staff, Laureldale Laboratory, Bell Telephone Laboratories, Inc., Laureldale, Pa.*

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

CONTENTS

10.1. Definition of Electrical Terms. 10–4
10.1.1. The Ohm 10–4
10.1.2. The Ampere 10–4
10.1.3. The Volt 10–4
10.1.4. The Coulomb 10–4
10.1.5. The Farad 10–4
10.1.6. The Joule 10–4
10.1.7. The Watt 10–5
10.1.8. The Henry 10–5
10.2. Relationships between Current, Voltage, Resistance, Capacitance, and Inductance 10–5
10.2.1. Ohm's Law 10–5
10.2.2. The Potential across an Inductance 10–5
10.2.3. The Potential across a Condenser 10–5
10.2.4. Kirchhoff's Laws 10–5
10.2.5. Series and Parallel Circuits 10–6
10.2.6. Simple *RC* Circuits 10–6
10.2.7. *RC* Coupling Circuits 10–7
10.2.8. Differentiating Circuits 10–7
10.2.9. *RLC* Circuits 10–8
10.3. Resistors 10–8
10.3.1. Fixed-composition Resistors 10–8
10.3.2. Wirewound Power Resistors 10–10
10.3.3. Precision Resistors 10–10
10.4. Vacuum Tubes 10–10
10.4.1. Vacuum Diodes 10–10
10.4.2. Triodes 10–11
10.4.3. Pentodes 10–12
10.4.4. Beam Power Tubes 10–14
10.4.5. Tube Parameters 10–15
10.5. Pulse Transformers 10–16
10.6. Delay Lines 10–17
10.6.1. Electrical Delay Lines 10–17
10.7. Magnetic Cores 10–18
10.7.1. Types of Cores 10–18
10.7.2. Bimags as Circuit Elements 10–19
10.7.3. Technical Data for Circuit Design 10–20
10.7.4. Core Constants 10–21
10.7.5. Noise in Bistable Magnetic Cores 10–21
10.7.6. Ferrite Memory Cores 10–21
10.7.7. Temperature Stability 10–22
10.8. Semiconductor Diodes 10–22
10.8.1. Introduction 10–22
10.8.2. Basic Processes 10–23
10.8.3. Rectification 10–23
10.8.4. Transient Behavior 10–25
10.8.5. Characterization and Test 10–25
10.8.6. OFF Impedance 10–26
10.8.7. ON Impedance 10–26
10.8.8. Switching Times 10–26

* Deceased.

10.8.9. Point-contact Diodes... 10–27
10.8.10. Alloyed-junction Di des. 10–28
10.8.11. Diffused-junction Diodes 10–29
10.8.12. Typical Characteristics.. 10–29
10.9. Transistors..... 10–29
10.9.1. Types of Transistors ($\alpha < 1, \alpha > 1$)..... 10–30
10.9.2. General Description..... 10–30
10.9.3. Symbolic Representation 10–30
10.9.4. Reverse Cutoff Currents. 10–30
10.9.5. Emission Efficiency γ.... 10–31
10.9.6. Transport Factor β..... 10–31
10.9.7. Collection Factor α^*.... 10–31
10.9.8. Frequency Cutoff of Alpha..... 10–31
10.9.9. Junction Capacitance.... 10–31
10.9.10. Base Resistance..... 10–32
10.9.11. Emitter and Collector Body Resistance..... 10–32
10.9.12. Avalanche Multiplication 10–32
10.9.13. Characterization..... 10–33
10.9.14. Large-signal Static Characterization..... 10–33
(*a*) Static Characteristics..... 10–33
(*b*) Regions of Operation..... 10–35
(*c*) Representation for D-C Behavior.... 10–35
(*d*) Negative-resistance Characteristics.... 10–37
10.9.15. Small-signal Transient Characterization..... 10–38
10.9.16. Large-signal Transient Characterization..... 10–39
10.9.17. Methods of Measurement and Test..... 10–41
10.9.18. Large-signal Measurements..... 10–41
10.9.19. D-C Measurements..... 10–41
10.9.20. Transient Measurements 10–45
10.9.21. Small-signal Measurements..... 10–45
10.9.22. Device Ratings and Rating Systems..... 10–46
10.9.23. Thermal or Power Ratings..... 10–47
10.9.24. Electrical Ratings..... 10–47
10.9.25. Mechanical Ratings..... 10–48
10.9.26. Discussion and Comparison of Different Types of Transistors..... 10–48
10.9.27. "Alpha Less Than Unity" Transistors..... 10–48
10.9.28. "Alpha Greater Than Unity" Transistors..... 10–50
10.9.29. Typical Transistor Characteristics..... 10–52
10.9.30. Variation of Parameters. 10–52
10.9.31. Reliability..... 10–54

10.1. DEFINITION OF ELECTRICAL TERMS

Of the various systems of electrical units in use throughout the world only one will be defined here. This is the one in most general use which was established by an International Electrical Congress in 1893 at Chicago.

10.1.1. The Ohm. One *ohm* is the unit of resistance to the flow of electric current and is represented by the resistance offered to a constant electric current flowing in a column of mercury 14.4521 grams in mass, of constant cross-sectional area, and of length 106.3 centimeters. The symbol Ω stands for one ohm. One kilohm equals 1,000Ω and usually is written as 1 K. $10^6\Omega$ is called one megohm and is frequently written as 1 M. Thus, 2.2 M = 2,200 K = $2.2 \times 10^6\Omega$.

10.1.2. The Ampere. One *ampere* of current is the equivalent of that constant current which, when in a solution of silver nitrate in water in accordance with standard specifications, deposits silver at the rate of 0.001118 gram per second. 10^{-3} ampere, written 10^{-3} amp, is called one milliampere and written 1 ma. 10^{-6} amp equals one microampere, written 1 μa. Thus, 0.01 amp = 10 ma = 10,000 μa.

10.1.3. The Volt. One *volt* is that electrical potential which when applied to a resistance of one ohm causes a current of one ampere to flow.

10.1.4. The Coulomb. One *coulomb* is the quantity of electricity transferred by a current of one ampere flowing for one second.

10.1.5. The Farad. One *farad* is a unit of capacity. A condenser which can be charged to a potential of one volt by one coulomb of electricity has a capacity of one farad. In electronic circuitry condensers usually range in size from about five micromicrofarads (5×10^{-12} farad) to perhaps 1,000 microfarads (10^{-3} farad).

10.1.6. The Joule. One *joule* is the unit of work which is equivalent to the energy expended in one second by one ampere in a resistance of one ohm.

10.1.7. The Watt. One *watt* is a unit of power equal to the work done at the rate of one joule per second. A watthour is 3,600 joules.

10.1.8. The Henry. One *henry* is a unit of induction which is the induction in a circuit across which a potential of one volt is developed by a change in current of one ampere per second. 1 henry = 1,000 millihenrys = 10^6 microhenrys.

10.2. RELATIONSHIPS BETWEEN CURRENT, VOLTAGE, RESISTANCE, CAPACITANCE, AND INDUCTANCE

10.2.1. Ohm's Law. If a potential is applied to a pure resistance current flows in the resistance in such a way as to satisfy Ohm's law:

$$e(t) = i(t) \cdot r(t)$$

where the units are volts = amperes · ohms, and as indicated in the equation the quantities may be functions of time. See Fig. 10.1.

10.2.2. The Potential across an Inductance. The potential across an inductance depends upon the change in the rate of flow of current according to the following relation:

$$e(t) = L(t) \cdot i'(t)$$

where the prime denotes differentiation with respect to time and the units are

volts = henrys · amperes per second

See Fig. 10.2.

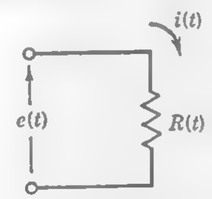


FIG. 10.1

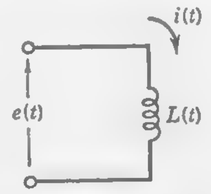


FIG. 10.2

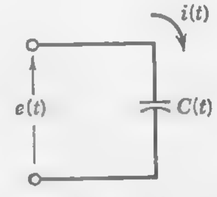


FIG. 10.3

10.2.3. The Potential across a Condenser. The potential across a condenser depends upon the charge and is given by the following equation:

$$e(t) = e(0) + \int_0^t \frac{i(t)}{C(t)}\, dt$$

where the potential is measured in volts, the capacitance in farads, the current in amperes, and time in seconds. In most applications the capacity is constant and the above equation can be written

$$e(t) = e(0) + \frac{1}{C}\int_0^t i(t)\, dt$$

The charge in coulombs is given by

$$Q(t) = C \cdot e(t) = C \cdot e(0) + \int_0^t i(t)\, dt$$

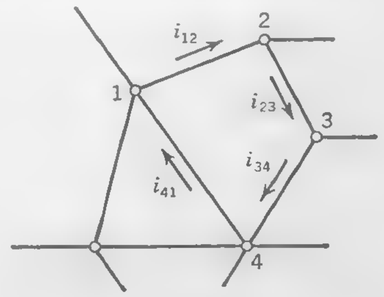


FIG. 10.4

10.2.4. Kirchhoff's Laws. Whenever circuit elements such as those described above are connected together to form a network the relations between currents and potentials in the branches of the network are described by Kirchhoff's laws. See Fig. 10.4.

1. The algebraic sum of the currents flowing toward any junction is zero.
2. The algebraic sum of the voltages around any closed path is zero.

Letters and arrows may be used to represent unknown currents in the branches of the network, and the direction of the arrows may be assigned arbitrarily. Negative currents flow in directions opposite to that indicated by the arrows. The signs of potentials on the various branches should be consistent with the directions indicated by the arrows.

10.2.5. Series and Parallel Circuits. Resistances, condensers, and inductances may be connected together to form networks. Such elements may be connected in series as indicated in Fig. 10.5 or they may be connected in parallel as indicated in

FIG. 10.5

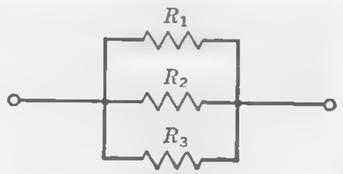


FIG. 10.6

Fig. 10.6. For resistances and capacitances connected in series and parallel the total resistance and total capacitance are given by the following formulas:

	Series	Parallel
Resistance.....	$R = \sum_i R_i$	$\frac{1}{R} = \sum_i \frac{1}{R_i}$
Capacitance.....	$\frac{1}{C} = \sum_i \frac{1}{C_i}$	$C = \sum_i C_i$

10.2.6. Simple *RC* Circuits. Suppose a resistance and a condenser are connected as indicated in Fig. 10.7. If the switch has been in the illustrated position for a relatively long time the potential $e(t)$ will be zero. If the switch is changed a current $i(t)$

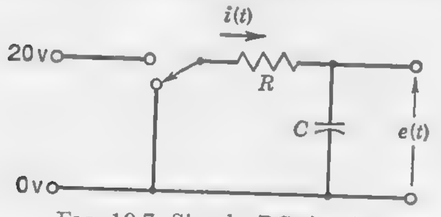


FIG. 10.7. Simple *RC* circuit.

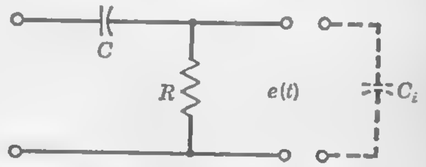


FIG. 10.8. *RC* coupling circuit.

flows through the resistance R and charges the capacitance C. Applying Kirchhoff's laws the current must satisfy the following equation:

$$e(0) + \frac{1}{C}\int_0^t i(t)\,dt + R \cdot i(t) = 20 \text{ volts}$$

This equation may be differentiated once to produce $i(t)/C + R \cdot i'(t) = 0$. Standard techniques give a solution of this equation of the form $a \cdot e^{bt}$ where $b = -1/RC$ and a is a constant determined by the boundary condition in the problem, namely, that the initial current $i(0+) = 20/R$. Integrating $(20/R)e^{bt}$ and taking into account that $e(0) = 0$ gives

$$e(t) = \frac{1}{C}\int_0^t \frac{20}{R} e^{bt}\,dt = 20(1 - e^{bt}) \qquad b = -\frac{1}{RC}$$

Inspection of the quantity $1 - e^{bt}$ shows that the way the potential approaches its final value depends upon the quantity $b = -1/RC$. $1/RC$ is referred to as the *time constant* of the circuit. Percentage of final value in terms of time constant is given in Table 10.1. Thus, in one time constant the output voltage has reached 63 per cent of its final value and in three time constants it has reached 95 per cent of its final value.

Table 10.1. Response Times

x	$1 - e^{-x}$
0.0	0.00
0.5	0.39
1.0	0.63
1.5	0.78
2.0	0.86
2.5	0.92
3.0	0.95
3.5	0.97
4.0	0.98

10.2.7. *RC* Coupling Circuits. Arrangements such as that illustrated in Fig. 10.8 are used to couple circuits where signals of short duration are developed at a different potential level from that at which they are to be used. By methods such as those illustrated above the response to a step voltage of 20 volts can be found to be

$$e(t) = 20e^{bt} \qquad b = \frac{-1}{RC} \qquad t \geq 0$$

Thus, if the input signal is a 20-volt pulse whose width is one time constant it is seen that the output potential is only 37 per cent of the input signal at the end of the pulse. Inspection of Table 10.2 shows that even if the coupling time constant is ten times the longest pulse to be transmitted the "droop" may be as much as 10 per cent.

Table 10.2. Coupling Constants

x	e^{-x}
0.0	1.00
0.1	0.90
0.2	0.82
0.3	0.74
0.4	0.67
0.5	0.61
0.6	0.55
0.7	0.50
0.8	0.45
0.9	0.41
1.0	0.37

In practice the input capacity of the circuit that is driven by the coupling circuit of Fig. 10.8 may not be negligible compared with C. In this case there is a capacitive division of the signal and the initial value of $e(t)$ is something less than 100 per cent. Suppose that C_i in Fig. 10.8 is 10 per cent of C, then the input potential, 20 volts, appears across the two condensers in series with the potential across each one proportional to the reciprocal of the capacity. Therefore, initially, approximately 18.2 volts will appear across C and 1.8 volts will appear across C_i. All the other values of Table 10.2 will be changed accordingly.

If the output of the coupling circuit is the grid of a triode the capacity C_i may not be constant. This effect becomes more important at high frequencies and will produce variations from the figures given in Table 10.2.

10.2.8. Differentiating Circuits. Very short time constants may be used in coupling circuits such as those described in the previous section. Then the output signal

is a pulse at the beginning and at the end of the longer signal as shown in Fig. 10.9.

10.2.9. *RLC* **Circuits.** Three components may be connected in a number of ways. Figure 10.10 shows one way of connecting a resistance, a condenser, and an inductance. This is an example of a "series-peaking" circuit used to improve the frequency response of amplifiers.

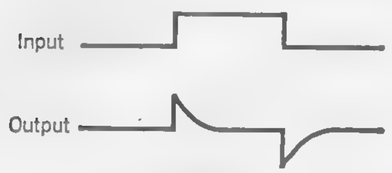


FIG. 10.9

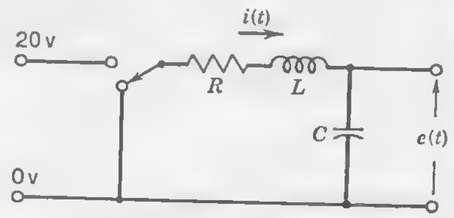


FIG. 10.10. *RLC* circuit.

The potentials and currents in this circuit may be represented by one of the following equivalent equations:

$$Ri(t) + Li'(t) + \frac{1}{C}\int_0^t i(t)\,dt = 20$$

or

$$LCi''(t) + RCi'(t) + i(t) = 0$$

Solutions of this may be written in the form $a \cdot e^{bt}$ where b is a solution of the equation

$$LCb^2 + RCb + 1 = 0$$

If the roots of this quadratic equation are complex the solutions are of the form

$$i(t) = e^{-Rt/2L} \sin \left(\frac{-R^2}{4L^2} + \frac{1}{LC}\right)^{1/2} t$$

10.3. RESISTORS

Resistors, condensers, and inductances or transformers are components which are used in almost all computer circuitry. Vacuum tubes have been extensively used in the past but now are being replaced by transistors and magnetic amplifiers.

In digital computers a single transient failure may void an extensive computation; so the practice is to design all circuits so as to operate with substantial margins. Most computers have means of checking that the circuits have not become marginal with age. Because of reliability requirements and the cost of maintenance most computer circuitry has been made with the best available components.

10.3.1. Fixed-composition Resistors. Fixed-composition resistors are used extensively in computer circuitry because of their small physical size, and for their good

Table 10.3. Resistor Color Code

Digit	*Color*
0	Black
1	Brown
2	Red
3	Orange
4	Yellow
5	Green
6	Blue
7	Violet
8	Gray
9	White
20%	No color
10%	Silver
5%	Gold

high-frequency response. Their disadvantages are lack of stability and noise. In high-signal-level pulse circuitry the noise is no problem.

Power Dissipation and Tolerance. Composition resistors come in sizes of ¼ to about 5 watts but are not extensively used in sizes above 2 watts. Because of lack of stability they are normally supplied with tolerances of 5 and 10 per cent.

Marking. A standard color code is used to mark composition resistors. Ten per cent resistors are marked with three color bands, two specifying the value of the resistance according to Table 10.3, and a third band specifying a multiplier as a power of 10. For example, a resistor of 270,000 ohms, or 270 kilohms, has three bands, which are red, violet, and yellow in order. A fourth band of silver indicating 10 per cent tolerance may be present.

Composition resistors are made in standard values, the difference between values being approximately twice the tolerance. Thus any value that is required in a circuit may be found in the appropriate tolerance group. Table 10.4 gives the preferred values for 5, 10, and 20 per cent resistors. Other preferred values are these multiplied by the appropriate power of 10. Resistors are regularly available in sizes from 10 ohms to 20 megohms. Many manufacturers sort their resistors so frequently a batch of 10 per cent resistors will not have many resistors within 5 per cent of the nominal value.

Table 10.4. Resistor Values

20%	10%	5%
10	10	10
		11
	12	12
		13
15	15	15
		16
	18	18
		20
22	22	22
		24
	27	27
		30
33	33	33
		36
	39	39
		43
47	47	47
		51
	56	56
		62
68	68	68
		75
	82	82
		91
100	100	100

Power Rating. The voltage across resistors is limited to that which will not exceed the dissipation rating or which will not exceed the dielectric strength of the material. The limits for most composition resistors are 350 volts for ½-watt resistors and 500 volts for 1- and 2-watt resistors. For heat dissipation the formula

$$P = \frac{E^2}{R} = I^2R$$

determines the size of the resistor to use. In warm environments resistors should be derated. For example, at 70°C they should be used at only 50 per cent of their rated load. For other temperatures the derating curve is linear.

10.3.2. Wirewound Power Resistors. Wirewound resistors are available in sizes ranging up to about 100,000 ohms and are capable of dissipating up to 200 watts of power. A typical range of values is 5, 10, 20, 25, 50, 80, 100, 120, 160, and 200 watts. The heat dissipation depends upon the physical size of the resistor. The size of a set of typical resistors is given in Table 10.5.

Table 10.5. Size of Wirewound Resistors

Size, watts	Length, in.	Diam, in.
5	1	5/16
10	1.75	5/16
20	2	7/16
25	2	5/8
50	4.5	3/4
80	6.5	3/4
100	6.5	1 1/8
160	8.5	1 1/8
200	10.5	1 1/8

Wirewound resistors may have ferrule, stack, tab, screw, or axial mounting arrangements. Usually they are made by winding resistance wire on a cylinder and protecting it with a plastic material or baked-on enamel. These resistors are more rugged and more stable than the composition types. They all have considerable inductive and capacitive effects. Even in the "noninductive" types these effects are not negligible at high frequencies. The typical behavior of a 10-watt, 10,000-ohm resistor may be represented by an equivalent circuit of Fig. 10.10 in which the inductance is somewhat more than 100 microhenrys (10^{-4} henrys) and the capacitance is less than 1 micro-microfarad (less than 10^{-12} farad). Thus, in some applications wirewound resistors may be used to advantage because of their series-peaking effect (see Art. 10.2.9).

10.3.3. Precision Resistors. For applications where high accuracy and stability are required and power dissipation is very low, wirewound resistors may be used. Deposited-carbon and metallic-film resistors may also be used in these applications. Resistance tolerances of 0.5 per cent are standard. Wirewound resistors should not exceed 105°C in these applications.

Metallic-film resistors are available in ½-, 1-, 2-, and 5-watt sizes. Both radial and pigtail leads can be had. Resistance sizes range from 1 ohm to 5, 10, 20, and 30 megohms (10^6 ohms). Deposited-carbon resistors are available in substantially the same styles and sizes as the metallic-film resistors.

10.4. VACUUM TUBES

Vacuum tubes come in a wide range of sizes which can best be characterized by wattage dissipation. Those tubes used in computer applications fall in the range from 1- to perhaps 50-watt size. Tubes used in computer applications may be classified as diodes, triodes, tetrodes, pentodes, and heptodes. All these tubes have a heater and cathode structure and an anode. The cathode is covered with photoemissive material such as barium oxide. If the anode is relatively positive with respect to the cathode the electrons emitted by the cathode are attracted to the anode, and anode current is said to flow. Diodes have only the heater-cathode and anode structures. Triodes have a third structure called the grid which can control the flow of current; tetrodes and pentodes have four and five active elements, respectively.

10.4.1. Vacuum Diodes. Diodes behave like rectifiers in that they allow the current to flow in only one direction. If the anode is positive with respect to the cathode electrons are attracted to the anode from the cathode. If the anode is negative relative to the cathode the electrons emitted by the cathode are repelled by the

anode. Figure 10.11 gives the relation between anode (or plate) voltage and anode current.

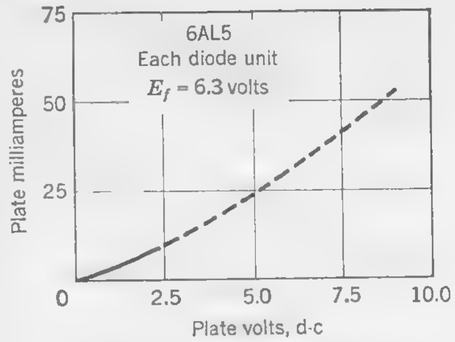


FIG. 10.11. Diode characteristic.

10.4.2. Triodes. The triode is similar to the diode except that a small wire has been placed between the cathode and the anode. The structure is so arranged that, if this grid is sufficiently negative in potential, effectively no electrons will flow from the cathode to the anode. If the grid is raised to the cathode potential then electrons do flow. In most tubes the grid cannot be raised above cathode potential except for extremely short intervals of time. The reason for this is that the grid structure must be extremely fine so as not to mask the flow of electrons from the cathode to the plate and yet it must effectively cover all the area between the cathode and anode in order to be able to control the flow of electrons. Thus, since it is made of fine wire, it cannot be raised above cathode potential or the electrons will be attracted to it and the generated heat would destroy the grid structure.

Figure 10.13 shows the relation of plate current to grid potential in a triode for constant plate voltage. Usually circuits are arranged so that the plate potential depends on the plate current, and then it is more useful to use Fig. 10.12, which gives plate current in terms of plate voltage at various grid potentials.

The size and arrangement of the electrode structures determine how much heat they can dissipate, and this determines the maximum potentials and currents at which the tube can be used.

Figure 10.14 gives the average plate characteristics of the GL6463 twin triode. Curves are given for grid potentials as high as plus 20 volts. The range of values given in Fig. 10.12 indicates the area of interest in designing amplifiers and other high-duty cycle circuits, whereas Fig. 10.14 gives a range of values which is very useful in pulse circuitry where the duty cycle is usually very low. Note that the grid current increases very fast with positive grid potentials, as indicated by the dashed lines of Fig. 10.14. As long as the average grid and plate dissipation are below the maximum

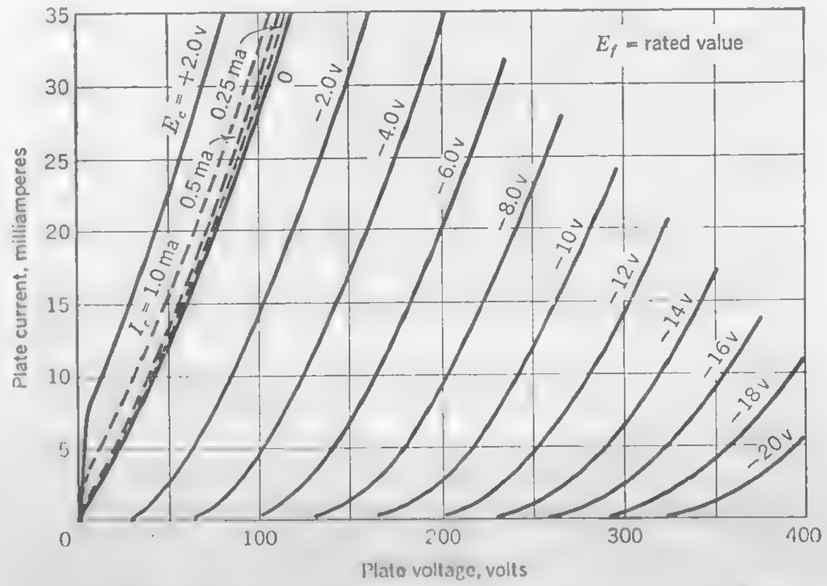


FIG. 10.12. GL 6463 characteristics, I_p vs. E_p. *(Courtesy of General Electric Co.)*

the tube will function properly. There are thermal time constants which limit the time that the grid may be in the region of positive potential. Note that a circuit operating at 125-kc pulse-repetition rate and using ½-μsec pulses has a duty cycle of less than 7 per cent.

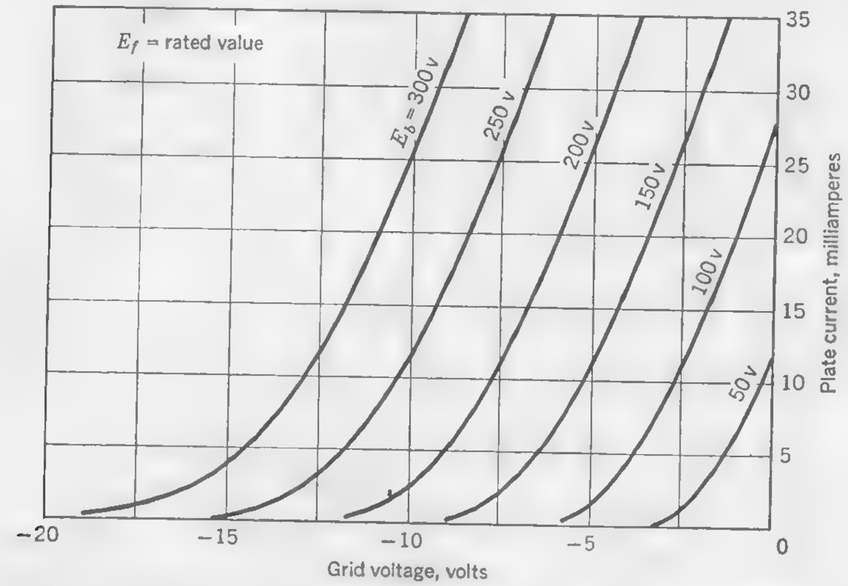


FIG. 10.13. GL 6463 characteristics, I_p vs. E_g. (*Courtesy of General Electric Co.*)

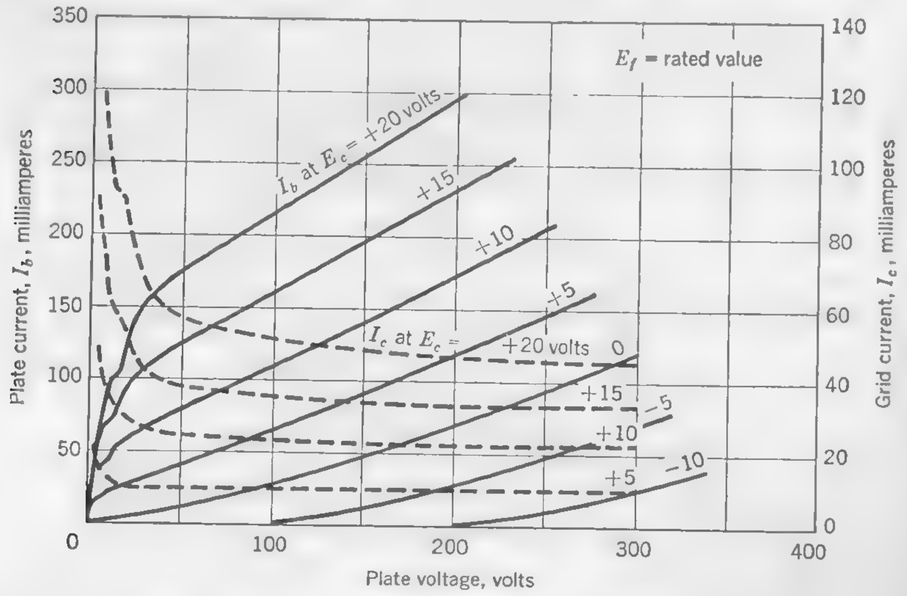


FIG. 10.14. GL 6463 characteristics for positive grid potentials. (*Courtesy of General Electric Co.*)

The 5687 has been used in computer circuits. Its characteristics are given in Figs. 10.15 and 10.16.

10.4.3. Pentodes. As was seen in Fig. 10.16 the plate current in a triode depends upon the plate potential. In the pentode this dependence has essentially been eliminated except for very low plate potentials. The pentode has five elements in the tube, namely, the cathode, grid, screen grid, suppressor grid, and plate. In normal operation the screen is kept at a fixed high potential and the suppressor grid is at cathode potential. As the name suggests, the screen grid shields the plate from the grid and cathode, making the cathode current nearly independent of plate potential. On the other hand, there are sufficient openings in the screen grid so that most of the electrons pass through to the plate.

When electrons strike a target secondary electrons are created. The suppressor grid is held at ground potential in order to prevent the exchange of such secondary electrons between the screen and plate. Usually it is a widely spaced structure that does not interfere with the flow of electrons from the cathode to the plate.

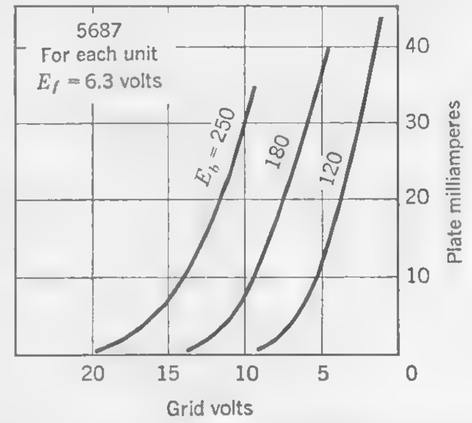


FIG. 10.15. (*Courtesy of Tungsol Lamp Works, Inc.*)

Figure 10.17 shows the plate current and screen current of a 6AN5 pentode as a function of plate voltage and control-grid voltage. Note that the curves are much flatter than they are for triodes. The 6AN5 is a high-current sharp-cutoff pentode developed particularly for computer applications. In contrast to other high-current pentodes substantial currents are available in the 6AN5 at low plate potentials and screen potentials. For example, at 100 volts on the plate, 120 on the screen, zero bias gives more than 90 mils of plate current, and with the grid reduced to −10 volt this plate current is reduced 90 per cent.

A number of multiple-element tubes or so-called dual-control tubes have found extensive application in computer circuitry. Examples are the 6AS6, GL5915A, and the 7AK7. The usual transfer characteristic and plate-current characteristic of the

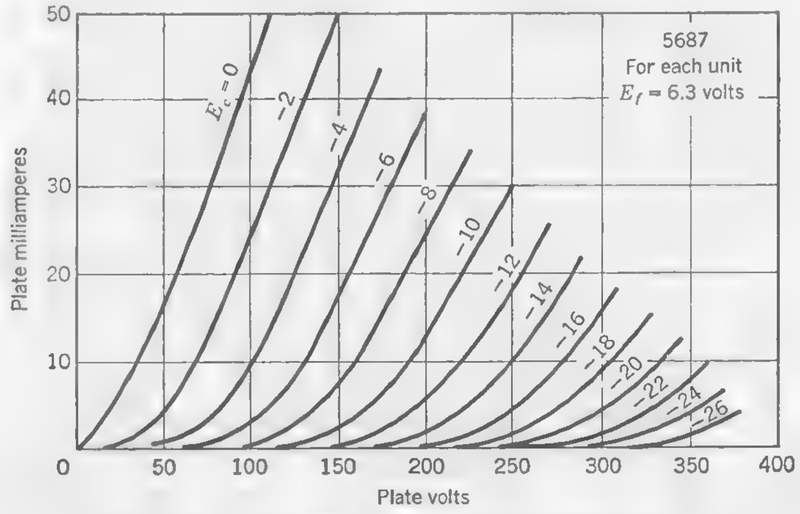


FIG. 10.16. (*Courtesy of Tungsol Lamp Works, Inc.*)

6AS6 are given in Figs. 10.18 and 10.19. Note that this is a low-current tube and that the screen current is higher than in most pentodes. The feature of real interest, however, is the control exercised by the suppressor grid (see Fig. 10.20). When the suppressor voltage is reduced to −10 volt the plate current has been reduced almost to zero (the screen current has increased). Thus plate current can be reduced by reducing the potential of the control grid or by reducing the potential of the suppressor grid. This makes this tube useful in the design of coincidence circuits.

The GL5915A is a heptode which is similar to the 6AS6 in performance. The 7AK7 was developed particularly for computer pulse operation and is a higher-current tube.

In most of these tubes the plate dissipation is higher than the screen dissipation; so circuit designers usually place the lower-duty-cycle signals on the control grid, and gating or switching signals on the suppressor grid.

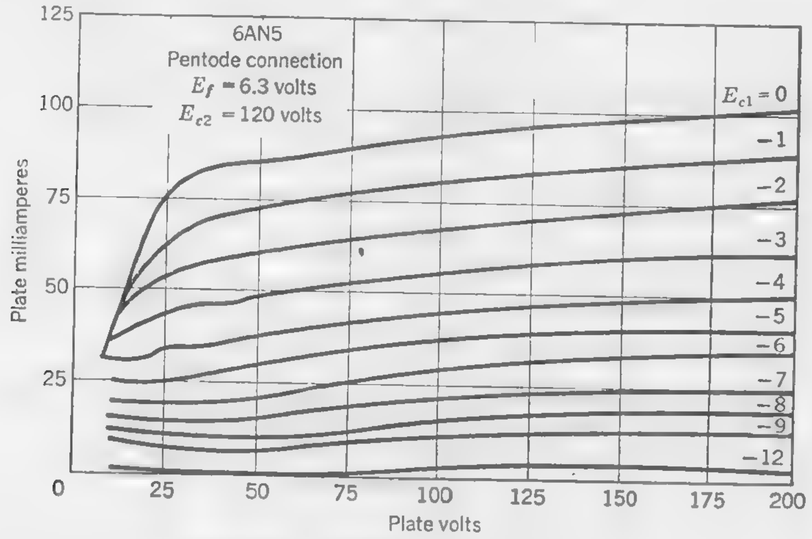


FIG. 10.17. (*Courtesy of Raytheon Co.*)

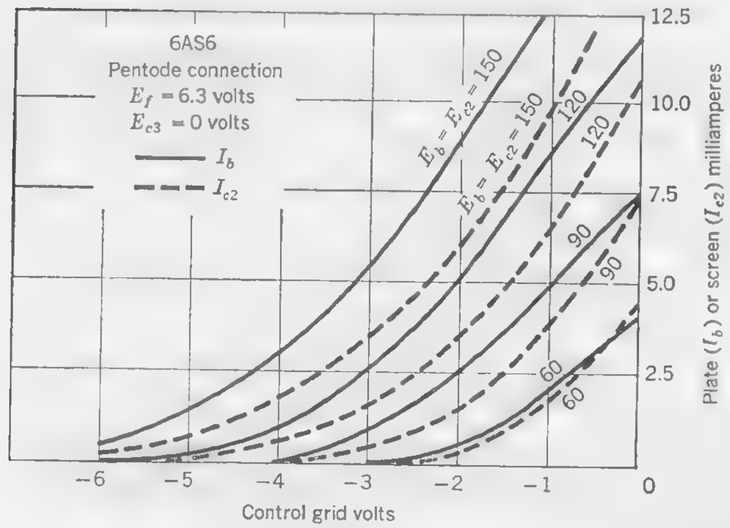


FIG. 10.18. (*Courtesy of Tungsol Lamp Works, Inc.*)

10.4.4. Beam Power Tubes. A beam power tube has four electrodes, and the effect of the suppressor grid is achieved by the arrangement of the other electrodes. The screen grid is arranged so that it lies in the shadow of the control grid, with the effect that there is less screen current than would be expected on the basis of intercepted area. The characteristics show no secondary emission effects except at very low plate voltages. The plate current raises very rapidly with plate voltage and then flattens off very sharply. Thus a beam power tube has a much sharper "knee" in the characteristic curves than pentodes do.

10.4.5. Tube Parameters. The tube parameters of most use in the computing field are the transconductance or g_m, zero bias plate current, and cutoff current. In communication applications transconductance is the most important parameter, and in digital-computer applications this parameter is of importance only in voltage-amplifier

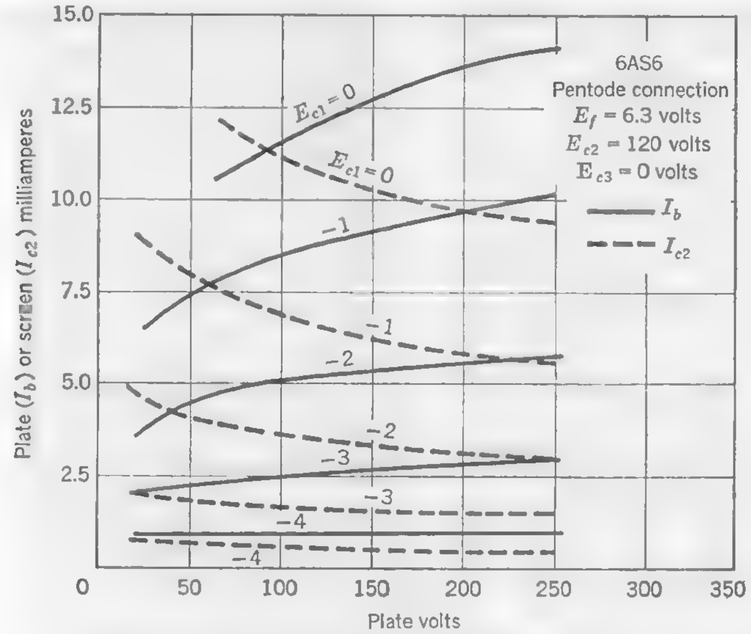


FIG. 10.19. (*Courtesy of Tungsol Lamp Works, Inc.*)

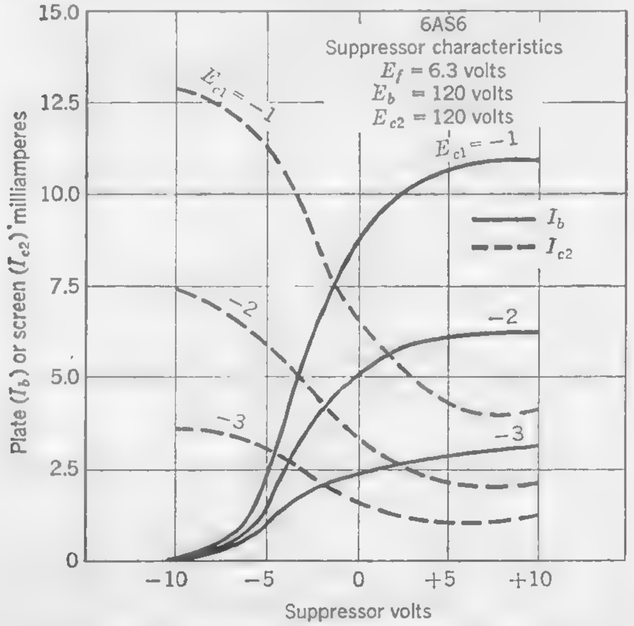


FIG. 10.20. (*Courtesy of Tungsol Lamp Works, Inc.*)

components. In other pulse-circuit applications zero bias current and cutoff are of more importance.

In pulse-amplifier circuitry or large-signal applications the zero bias current indicates whether or not the tube will produce a signal of adequate size. Cutoff current indicates the response of the circuit to noise. Transconductance may be correlated

with the rise of the leading edge of the pulse, but frequently this is of little importance in computer circuitry.

10.5. PULSE TRANSFORMERS

Pulse circuits may be interconnected by means of pulse transformers as well as by *RC* circuits (see Art. 10.2.7). The advantages of pulse-transformer coupling are that it is possible to match or change impedances and that the signal may be inverted. With more than two windings it is possible to connect together circuits that operate at three or more d-c levels. The disadvantages in transformer coupling are no d-c coupling, a possible change in base-line level of the pulses depending upon duty cycle, and frequency limitations due to recovery of the transformer.

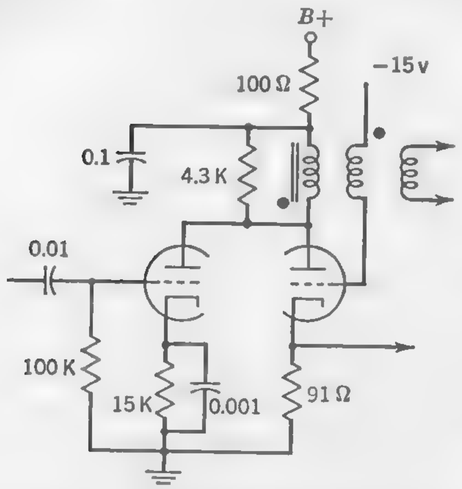


FIG. 10.21. Transformer test circuit. (*Courtesy of Sprague Electric Co.*)

A pulse transformer suitable for 1-μsec pulses may be made by winding a primary of 150 turns of No. 40 wire on a form ⅜ by ¾ in., covering this with five turns of 3-mil scotch tape, and then winding the secondary of such wire size and number of turns as necessary to cover the scotch tape. The core may consist of, say, 7 ft of 1-mil Hypersil threaded through the form. Terminals can be added and the whole protected by a casting resin.

An easier method of fabrication involves using pot cores (such as Ferroxcube 3C type 7F154) and winding the primary and secondary on a bobbin to be placed inside the core material. The primary should be spaced from the secondary either by a partition on the bobbin or by other insulating material between the windings.

Pulse transformers suitable for computer applications are available commercially for pulse widths from 0.02 to greater than 25 μsec and have been made with turns

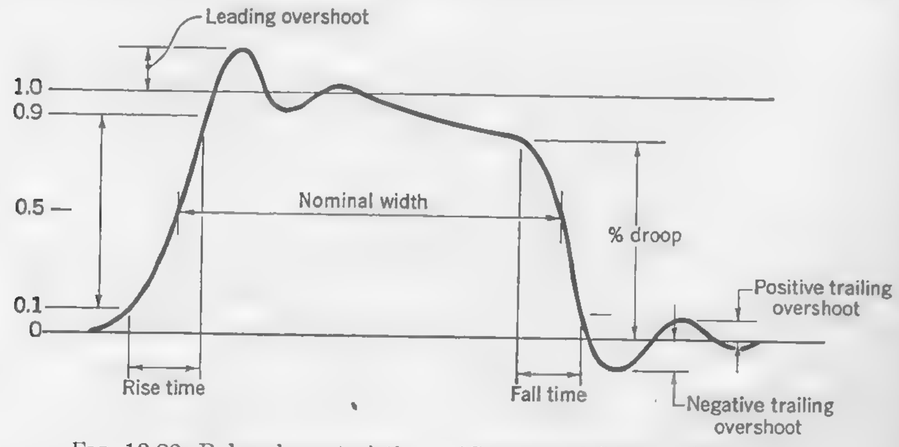


FIG. 10.22. Pulse characteristics. (*Courtesy of Sprague Electric Co.*)

ratios as high as 40:1. In some sizes, transformers with three and four windings are available.

The nominal pulse width of a pulse transformer is measured by observing its performance in a blocking-oscillator circuit such as that shown in Fig. 10.21.

Figure 10.22 shows how the various characteristics of a pulse are specified, and Table 10.6 gives some characteristics of typical pulse transformers used in computers.

Table 10.6. Typical Pulse-transformer Characteristics*

Nominal pulse width, μsec	Rise time, μsec	Fall time, μsec	% droop
0.40	≤0.05	≤0.2	≤10.0
0.65	0.05	0.2	10.0
1.25	0.05	0.2	10.0
2.00	0.05	0.2	10.0
2.50	0.05	0.2	10.0
0.5	≤0.05	≤0.3	≤6.0
0.8	0.05	0.3	6.0
1.1	0.05	0.3	6.0
1.5	0.05	0.3	6.0
1.9	0.05	0.3	6.0
2.2	0.05	0.3	6.0
2.6	0.05	0.3	6.0
2.0	≤0.15	≤0.6	≤10.0
2.6	0.15	0.6	10.0
3.4	0.15	0.6	10.0
4.2	0.15	0.6	10.0
4.9	0.15	0.6	10.0
5.5	0.15	0.6	10.0
6.0	0.15	0.6	10.0
6.4	0.15	0.6	10.0
5.2	≤0.35	≤1.5	≤14.0
7.5	0.35	1.5	14.0
10.0	0.35	1.5	14.0
12.3	0.35	1.5	14.0
14.4	0.35	1.5	14.0
15.8	0.35	1.5	14.0
17.0	0.35	1.5	14.0
17.6	0.35	1.5	14.0

* Courtesy of the Sprague Electric Co.

10.6. DELAY LINES

A delay line is a device such that if a signal is applied to the input it appears at the output at a fixed time later. The input may be pulses, modulated radio-frequency signals, or d-c signal changes. In this section electrical delay lines will be discussed which can pass all three kinds of signals. Acoustic lines will also be considered, but in some cases there are limitations in the kind of signal which can be passed through such a delay line.

10.6.1. Electrical Delay Lines. Electrical delay lines are of two types, lumped constant or continuous. The lumped constant consists of several small inductances connected in series with a capacitor connected to ground at each junction. The continuous line is the limiting case of the lumped-constant line in that there is one continuously wound inductance and the capacity is that between the windings of the inductance and a shield which is placed near the inductance. Some delay lines are of the air-core type; others use a magnetic material to increase the delay obtainable in a given physical dimension.

Continuous electrical delay lines are commercially available which range in impedance from 400 ohms to 4 kilohms, with delays of from 0.1 to 1 μsec per ft, and in bandwidths to 15 Mc. The attenuation averages around 2 to 3 db per μsec of delay. Table 10.7 gives the characteristics of typical lines.

Table 10.7. Electrical Delay Lines*

Length, in.	Delay, μsec	Rise time, μsec	Characteristic impedance, ohms	D-C resistance, ohms	Attenuation, %
1.5	0.1	0.025	150	3	2
1.5	0.25	0.06	500	4	1
1.5	0.5	0.12	700	7	1
1.5	1.0	0.24	1,400	45	3
1.5	2.0	0.5	2,000	70	3
1.5	3.0	0.75	3,000	100	3
3.0	0.25	0.035	500	8	2
3.0	0.5	0.065	200	6	4
3.0	1.0	0.15	400	16	3
3.0	2.0	0.28	1,400	90	6
3.0	6.0	0.65	3,000	200	6

* Courtesy of the Valor Electronics Co.

10.7. MAGNETIC CORES*

10.7.1. Types of Cores. Magnetic cores have been developed in two basic forms, the metal-strip types and the molded ferrites. The strip cores usually consist of a number of wraps of a microthin alloy tape wound on a ceramic bobbin, spot-welded in place and annealed. They may be plastic-covered, or potted in resin to afford greater mechanical and chemical protection. Ranging in nominal sizes from about ⅛ to 1 in. diameter, metal cores are generally employed in manipulative or logical types of circuits. Although considerably more expensive than ferrite cores, they are less affected by temperature change and require less power under comparable operating conditions.

The ferrites are fabricated from various mixtures of pure metals, metallic oxides, and binding materials pressed into toroid shapes and fired at high temperatures—a process familiar to the ceramics industry. They may be obtained in sizes smaller than is possible with wound strip, diameters of 80 mils (0.080 in.) being quite common. Because of size and low cost, the ferrites are generally preferred in static-memory applications where the number of cores required is apt to be excessive.

Irrespective of form, cores are further classified as square or nonsquare in accordance with certain specific properties of the special materials of which they are composed. The terms derive from the shapes of the hysteresis loops characteristic of the two separate classes of magnetic materials later to be described. In general, square cores are useful as bistable devices, whereas the nonsquare cores are perhaps basically more conventional in function.† The phenomena that are of particular interest in circuit application are those which appear when a core is subjected to a magnetic field that is periodically reversed in direction. After a few reversals of this nature, the sample enters a cyclical condition in which the induced flux density B is determined by the magnetizing field H and its sense of alteration, as illustrated in the closed curves of Fig. 10.23.

The vast majority of materials used in components such as transformers and inductors of the more conventional types have S-shaped loops as shown at (*a*). The unique characteristic of the materials that have been developed for use in bimags (bistable magnetic cores) is that they exhibit a rectangular or square-shaped hysteresis curve of the form shown at (*b*). Here the coercive force H_c is but a small fraction of that required with ordinary magnetic materials. Another distinguishing property is that

* By Isaac L. Auerbach.

† While nonsquare cores are found in bistable ferroresonant combinations, they function as nonlinear inductances rather than bistable devices in themselves.

the residual flux density, or magnetization, is very nearly that of the maximum density. The squareness ratio B_r/B_m, with unity as the ideal value, ranges from about 0.85 to 0.95 in generally available and acceptable cores.

10.7.2. Bimags as Circuit Elements. Since bimags are ordinarily supplied without input or output windings, it is common practice to wind magnetic cores with copper

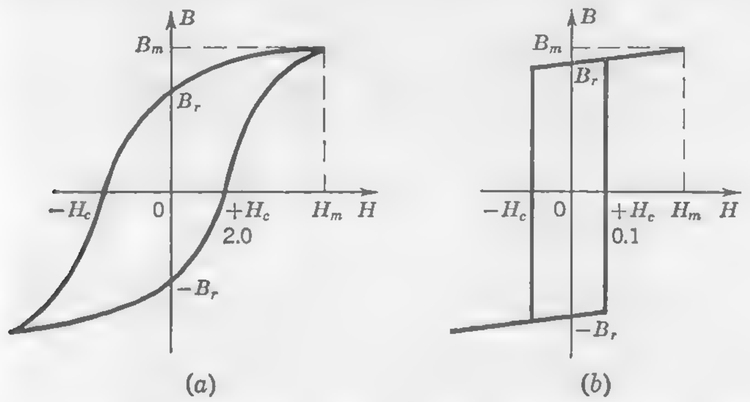


Fig. 10.23. Hysteresis loops associated with (*a*) most industrial magnetic materials, and (*b*) bistable magnetic cores.

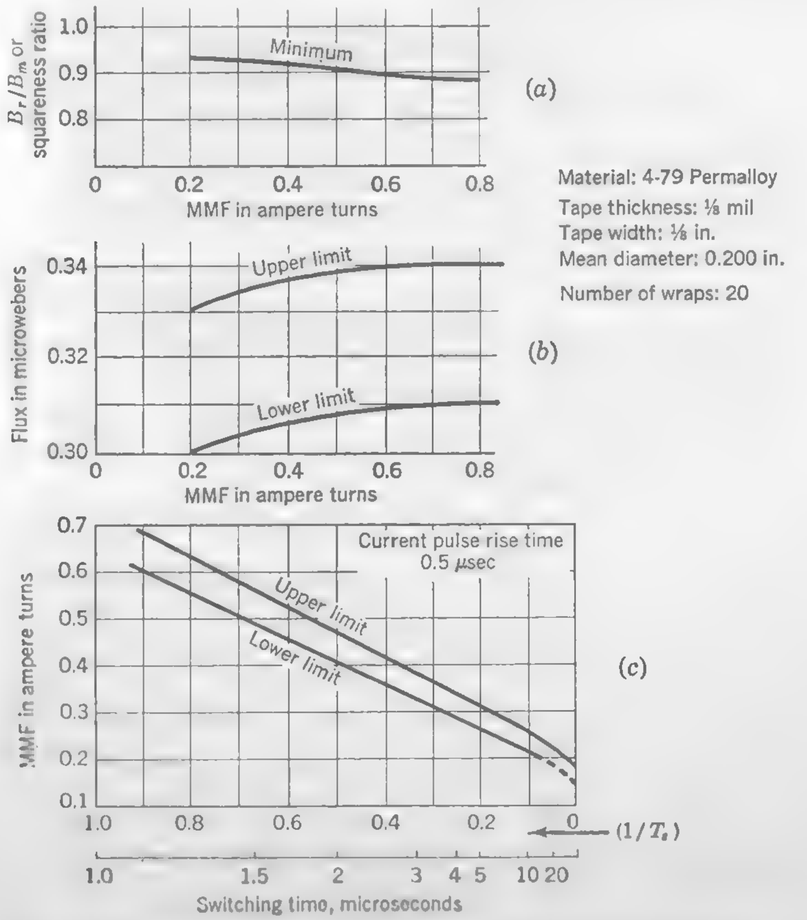


Fig. 10.21. Characteristic curves for a typical metal-strip magnetic core.

wire at circuit assembly* in conformity with design requirements. So fitted, the bimag becomes a circuit element in the more usual sense, but one which lends itself to flux-current considerations rather than voltage-resistance analysis.

The usefulness of the bistable magnetic core derives from the fact that, once the material is magnetized in one sense by current flow through a winding, it will remain so until magnetized in the opposite sense† by reversal of the current direction. The

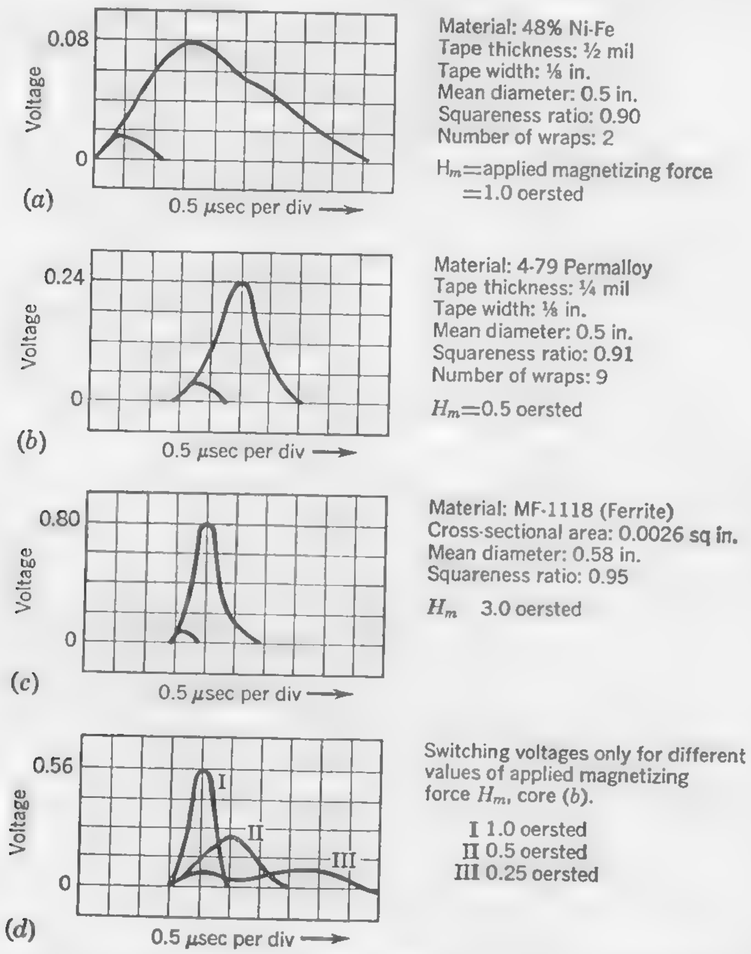


FIG. 10.25. Switching and nonswitching voltages of various bistable magnetic cores.

reapplication of magnetization in the same direction is accompanied by very little flux change, and hence negligible induced voltages in the various core windings. It is possible to distinguish between the two states of magnetization by applying an input current pulse and sensing the flux change by electrical means.‡

10.7.3. Technical Data for Circuit Design. While the basic properties of magnetic cores are described by the hysteresis curves, of more practical importance are the relationships between magnetomotive force NI and switching flux Φ_s, switching time T_s, and squareness ratio S. The curves are generally shown, in the case of Φ_s and T_s, as

* In certain applications, e.g., the coincident-current matrix memory, the smaller ferrite cores may simply be threaded upon the wires (see Art. 10.7.6).

† The change from one stable state to the other in which the flux direction is rapidly reversed is referred to as "switching" (see Art. 12.3.1).

‡ In this connection, the bimag may be treated as an ampere-turn transformer or, alternatively, as a bistable impedance.

upper and lower limits over the recommended operating range. Squareness ratio S is important as a figure of merit relating to the electrical quality of the particular core. The curves of Fig. 10.24 are typical and apply to a metal-strip core appropriate for use in manipulative-type circuits.

10.7.4. Core Constants. Two useful constants may be derived from the switching-time characteristic (Fig. 10.24*c*). The pulse-threshold magnetomotive force F_0, important from the standpoint of noise immunity and corresponding to the intercept at $1/T_s = 0$, is defined as that value of magnetomotive force below which no switching is possible. Coefficient G, the slope of the curve referred to the $1/T_s$ base, is constant over the useful operating range. With the aid of the upper and lower limits of Φ_s, F_0,

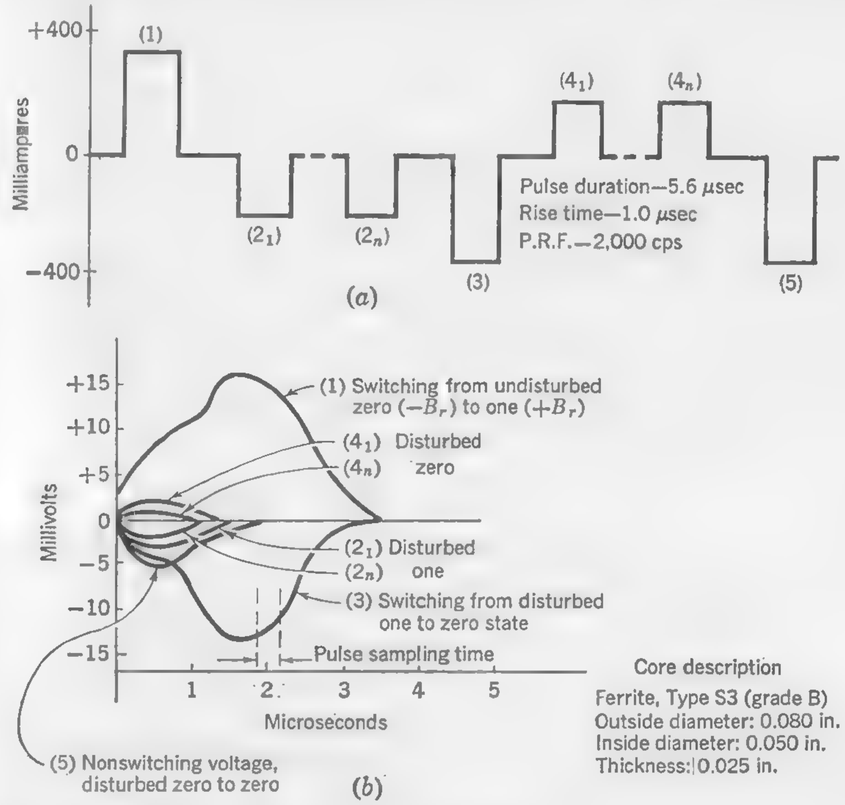


FIG. 10.26. Determination of minimum signal and maximum noise in the output of a ferrite core for memory-matrix application.

and G it is possible to make quantitative calculations of a core's performance under many conditions without resort to the complete characteristic curves.*

10.7.5. Noise in Bistable Magnetic Cores. A source of noise inherent in magnetic cores may be traced to nonsquareness in the hysteresis characteristics.† The amount of flux change produced when a core is driven from remanence to the corresponding point of saturation can be considered as false information. The squareness ratio S is significant in this connection since the noise source theoretically would vanish, were the ideal value of 1.0 attainable in presently available square-loop magnetic materials.

Figure 10.25 illustrates noise voltages observed in the test of three representative bistable cores. The information signals, present during switching of the cores from one stable state to the opposite state, are also shown for purposes of comparison.

10.7.6. Ferrite Memory Cores. Ferrite cores of special-grade material are generally required for coincident-current memory applications. Carefully manufactured

* See Art. 12.3.1.

† See Arts. 12.3.1 and 12.3.3 for a detailed discussion of this topic.

and tested for uniform characteristics, these minute cores change state in response to a current pulse of specified waveform but are materially unaffected by pulses of one-half the amplitude.

To evaluate the cores properly for the memory application, test specifications generally are formulated which subject a single core to conditions that might be encountered in an operating array. Figure 10.26 shows a current-pulse pattern which (1) sets a typical memory core in the positive state of magnetization, (2_1 . . . 2_n) disturbs the state by a series of partial pulses that tend to reduce the magnitude of residual flux, (3) switches the core from the disturbed positive state to the negative state, (4_1 . . . 4_n) disturbs the negative state toward the positive, and (5) restores the core from the disturbed negative state to the normal negative state. In the case of a satisfactory core, the amount of flux change during disturbance is not a function of the number of partial pulses, provided the number of such pulses is greater than some small number, usually two or three. The flux changes, moreover, should be as small as possible since the voltages induced during disturbance constitute additional noise in the sense described in Art. 10.7.5.

Signal-to-noise ratios in the outputs of individual cores accordingly are considerably less favorable, on a peak-amplitude basis, than might be expected in applications where partial pulses need not be taken into account. These ratios, however, usually may be improved by employing sampling methods in the matrix output equipment.

10.7.7. Temperature Stability. Normal changes in operating temperature have negligible effect upon the characteristics of metal-strip cores. Variations of less than ±2 per cent over the range 0 to 150°F are not uncommon in the case of many quality components. The molded ferrites, on the other hand, are considerably less stable, particularly the special grades supplied for coincident-current matrix applications. Reliable operation of these systems usually requires control of core temperatures within a few degrees of a specified ambient value.

10.8. SEMICONDUCTOR DIODES*

10.8.1. Introduction. Semiconductor diodes are used in switching or computing applications to perform many functions. Diode gate circuits, for example, consist of an assemblage of diodes, operating as switches, whose states can be altered in a prescribed manner by the application of input signals to provide an output signal. Diode switches are also used to limit the voltage swing of a given circuit node, or to clip or reshape pulses. The usefulness of diodes as switches is attributed to the high degree of nonlinearity of their voltage-current characteristic.

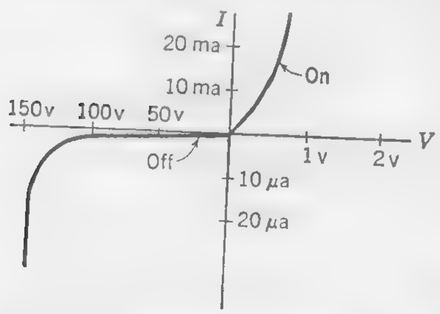


FIG. 10.27. Semiconductor-diode characteristics.

Both point-contact and junction diodes are used in computer applications. Point-contact diodes and small-area junction diodes are inherently fast and may be used in high-speed gate or logic circuits. Most of the present-day computer applications for large-area junction diodes are as rectifiers, voltage limiters, low- or medium-speed gate elements, or voltage-reference devices.

A typical semiconductor-diode characteristic is shown in Fig. 10.27. The ON condition of a diode switch corresponds to the diode being biased in the forward, low-impedance direction. The OFF condition corresponds to the diode being biased in the reverse, high-impedance direction. The important characteristics of a diode switch are the ON impedance, the OFF impedance, and the switching time from OFF to ON and from ON to OFF.

The *p-n-p-n*[46,†] and Esaki[36] types of diodes differ from the more conventional types

* By J. J. Ebers and M. L. Embree.

† Diode references are given in Art. 10.8.13.

in that they both exhibit negative-resistance characteristics. These types are particularly advantageous in special circuit applications where regenerative operation is utilized.

10.8.2. Basic Processes. Semiconductor junction diodes can be fabricated by several techniques. They can be grown into a crystal as it is withdrawn from the molten material. In this case the crystal may initially be grown one resistivity type, say *n* type, and then enough *p*-type impurities are added to overcompensate for the *n*-type impurities. Junctions may also be obtained by alloying and/or diffusion (see Art. 10.9.26). In surface-barrier transistors, rectifying contacts are obtained by plating a metal on a semiconductor surface; these rectifying contacts behave, in most respects, like *p-n* junctions. With surface treatments which tend to give an inversion

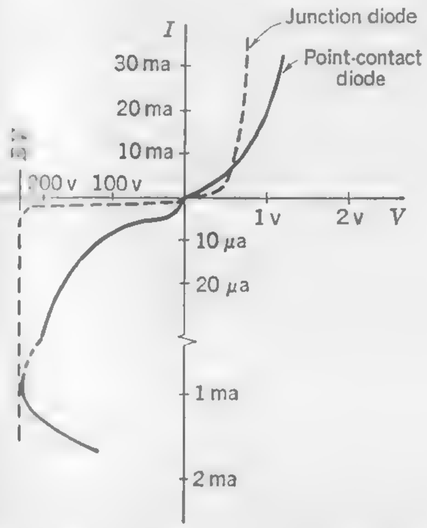


FIG. 10.28. Comparison of point-contact and junction diode characteristics.

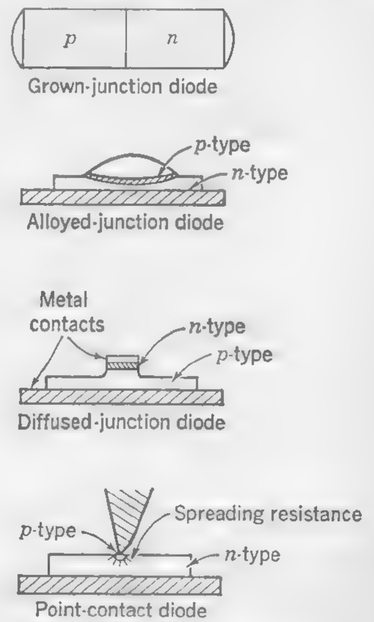


FIG. 10.29. Diode structures.

layer near the semiconductor surface, the semiconductor surface will rectify. In most point contact germanium diodes a true *p-n* junction is obtained in the semiconductor beneath the point by a process called forming. In the forming process a surge of electrical energy is used to heat the semiconductor and point. A possible result is that copper is diffused into the *n*-type germanium to form a *p*-type region under the point. An alternate explanation for the occurrence of the *p*-type region under the formed point is that during the forming process the semiconductor material undergoes plastic flow which introduces imperfections which behave as *p*-type impurity centers. Irrespective of the above and for purposes of understanding their behavior all semiconductor diodes might be considered to be comprised of a *p-n* junction.

10.8.3. Rectification. Typical rectification characteristics of junction and point-contact diodes are shown in Fig. 10.28. The theoretical relation between junction voltage and junction current is[62,25,26,52]

$$I = I_s \left(e^{\frac{qV}{kT}} - 1\right) \tag{10.8.1}$$

where I_s is the reverse saturation current and kT/q is a constant equal to $\simeq$0.026 volt at room temperature. The saturation current I_s is a parameter dependent upon the geometry and material constants of the diode as well as the surface conditions of the semiconductor material. It is a quantity which is quite temperature-dependent, doubling every 8°C in diodes which have close to theoretical reverse characteristics.

The saturation current results when minority carriers in the semiconductor regions adjacent to the junction drift over and are collected by the reverse-biased junction.

Equation (10.8.1) does not accurately express the current-voltage relationship in all diodes for several reasons. Many diodes, particularly point-contact diodes, frequently behave as if there were a leakage resistance in parallel with the junction which may range from thousands of ohms to many megohms. This leakage resistance is usually associated with the semiconductor surface. In some diodes an apparent leakage resistance can be attributed to avalanche multiplication of minority carriers.

The forward current of the semiconductor diode usually does not follow the rectification equation because of resistance in the semiconductor body. This resistance may be very significant in some point-contact-diode designs because of the spreading resistance under the point (see Fig. 10.29). This series resistance is also frequently a decreasing function of current because of an effect called conductivity modulation. Conductivity modulation results from the junction injecting minority carriers into the regions adjacent to the junction, for example, electrons into the p-type material. Additional holes in turn are required in order to maintain space-charge neutrality. Therefore, the conductivity of the p-type material is increased.

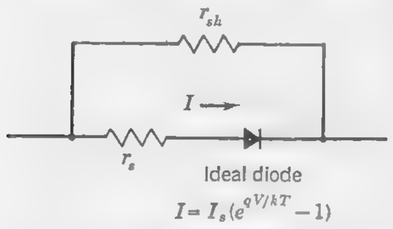


FIG. 10.30. Low-frequency equivalent circuit of semiconductor diode.

Because of leakage and series resistance, a suitable d-c equivalent circuit for semiconductor diodes is as shown in Fig. 10.30. This circuit will quite accurately represent the d-c behavior of diodes for reasonable values of current and voltage though the series and shunt resistances may show variation with operating point.

The slope of the forward characteristic, or the a-c resistance, is given by the following expression which results from differentiation of Eq. (10.8.1) and addition of r_s

$$z_E = \frac{kT}{qI} + r_s$$

where r_s is the series resistance and $kT/q \simeq 0.026$ volt at room temperature.

As the reverse voltage on a junction diode is increased, a voltage is eventually reached at which the reverse current begins to increase quite rapidly (see Fig. 10.28). This voltage has been called the breakdown voltage and is usually caused by a process called avalanche multiplication,[40,42] although Zener breakdown (internal-field emission)[39] may occur in some low-voltage junction diodes. The breakdown is normally nondestructive and may either occur within the body of the semiconductor junction or on the surface around the periphery of the junction. In some diodes, the leakage current may be so large as to mask the avalanche breakdown voltage; these diodes are said to have a "soft" reverse characteristic. Unless pulse-measurement techniques are used, point-contact diodes normally break down in a different manner. For these devices a reverse voltage is ultimately reached at which the reverse current is large enough to cause appreciable heating of the semiconductor beneath the point. This results in large numbers of available current carriers and the diode reverse current increases. Under these conditions, the increased current results in greater heating which results in more current, etc. This process represents a thermally induced negative-resistance characteristic (see Fig. 10.28) which, under certain circuit conditions, can result in diode destruction. Because of this phenomenon, it has been customary to define a peak inverse voltage (PIV) for point-contact diodes. This rating is usually specified by the manufacturer and should not be approached in normal operation.

The avalanche breakdown voltage (BV) of a junction diode has been found to increase with temperature.[60] This variation of breakdown voltage with temperature is less pronounced in low-voltage diodes (fabricated using low-resistivity material). The variation is approximately 0.08 per cent per °C for breakdown voltages of the order of 25 volts and 0.02 to 0.04 per cent for 6.8-volt diodes.

10.8.4. Transient Behavior. The transient or switching behavior of semiconductor diodes is affected by at least three factors: junction capacitance, storage of minority carriers, and conductivity modulation. If a voltage step function is applied to the diode in the forward direction, the current does not immediately reach its steady-state value. This is because it takes a finite time for injected minority carriers to reach an equilibrium density in the semiconductor regions adjacent to the junction and cause the decrease in semiconductor body resistance (conductivity modulation). Because of this conductivity modulation, the current will increase from some initial value to the steady-state value as shown in Fig. 10.31.

If a reverse bias is suddenly applied to a semiconductor diode, after the diode has been conducting forward current, a large reverse current will flow because of storage of minority carriers which were injected into the regions adjacent to the junction during the flow of forward current. Until the density of carriers at the junction essentially reaches zero, the junction retains a forward bias. After a period of time, called the storage time, the density of carriers near the junction approaches zero since they are being collected under the conditions of applied reverse bias and the reverse current returns toward its steady-state value. During the time that the current is decreasing toward its steady-state value, the current is comprised of minority carriers which have

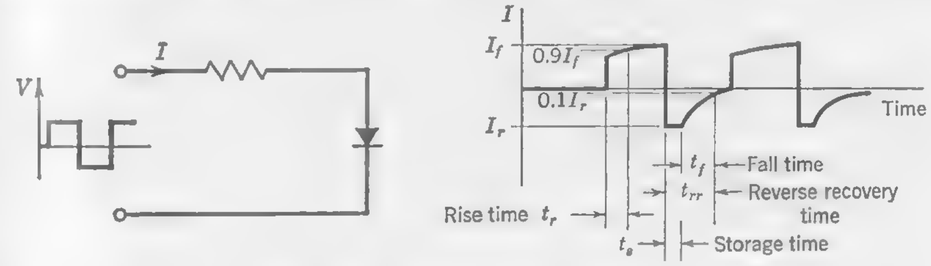


FIG. 10.31. Diode transient behavior.

been drifting toward the junction from the regions remote from the junction. Under some conditions, a significant portion of this current may also be that which is required to charge the junction capacitance. This effect is pronounced if the time constant of the circuit resistance and junction capacitance is long compared with the time constant for the transit of the minority carriers to the junction.

The storage time is dependent upon the design of the diode and upon the rate at which the minority carriers are withdrawn (see Fig. 10.31). One way in which diodes can be designed to have low storage times is to limit the size of the regions in which the minority carriers can be stored. This is done by placing the ohmic contacts as close as possible to the junction. Another method is to use semiconductor materials in which the minority carrier lifetime is low and hence many of the carriers die before they can be collected. Point-contact and other small-area diodes have low storage times because the current carriers injected under forward-bias conditions move away from the junction and spread out in a hemispherical manner. Because these carriers move largely by diffusion, only a small fraction can diffuse back to the junction under conditions of applied reverse bias.

10.8.5. Characterization and Test. The important characteristics of a diode switch are the ON impedance, the OFF impedance, and the switching times. The ON and OFF impedances are directly related to the rectification characteristic of the diode, and for some purposes it may be advisable to obtain a point-by-point plot of this characteristic. The rectification curve may also be traced out on an oscilloscope. For this visual method it may be advisable to use different scales for the forward and reverse characteristics by switching from one measuring circuit to another during alternate half cycles of the applied voltage. For acceptance or production testing, however, the ON and OFF impedances are usually specified in terms of a small number of well-chosen parameters (see Arts. 10.8.6 and 10.8.7).

The switching or transient behavior of a diode is expressed in terms of forward-current rise time, forward-peak-voltage transient, and reverse recovery time.[78,7,59,20]

Whereas it is theoretically possible to relate these times to basic diode parameters[31] (like alpha and alpha cutoff frequency in transistors), they are usually specified in terms of the diode behavior in a given circuit and under a given set of conditions.

10.8.6. OFF Impedance. The reverse characteristic of semiconductor diodes is usually specified in terms of the current at one or two values of voltage. The breakdown voltage of junction diodes is usually specified in terms of the voltage at a given current such as 10 or 100 μa. For some applications the slope of the reverse breakdown characteristic of junction diodes is important. This can be measured quite easily by biasing the diode in the breakdown region and measuring the a-c impedance. The value of voltage at two different values of current may be a preferable characterization.

Space-charge-layer capacitance[62] may have an appreciable effect on the OFF impedance of the diode switch, and since this capacitance is a function of the applied voltage, measurements are taken at a specified voltage. If some idea of the variation of capacitance with voltage is needed, one or more additional readings may be required. The capacitance will normally vary inversely as some power of the voltage, except for point-contact diodes in which the structural parts may contribute a large fraction of the total capacitance.

10.8.7. ON Impedance. The forward characteristic of semiconductor diodes may be specified in terms of the voltage drop at one or more values of forward current. The small-signal impedance at a fixed value of forward-bias current is frequently important in switching circuits and is also a measure of the body resistance of the diode. It may be necessary to make this measurement at more than one value of forward current because of the effect of conductivity modulation which decreases the body resistance as the forward current is increased. Forward-peak-voltage transient occurs (see Art. 10.8.8) because a finite time is required for the injected minority carriers to reach equilibrium density. During this period the effective series resistance of the diode decreases because of conductivity modulation.

10.8.8. Switching Times. The forward-current rise time is defined as the time required for the diode current to reach a given fraction, say 90 per cent, of its final value upon application of a step function of voltage. This time is largely independent of the value of reverse bias immediately before the application of the forward bias. The reverse recovery time of a diode is composed of two components, storage time and reverse-current fall time. One of the ways in which diode storage time and reverse-current fall time are defined can be explained by reference to Fig. 10.31. The diode is initially carrying a forward current I_F. A reverse-bias voltage is suddenly applied through a series resistance to cause a reverse current I_R to flow. For a period of time t_s called the storage time, the junction remains forward-biased and I_R is essentially the applied voltage divided by the series resistance of the circuit. At the end of the period of time t_s, the density of minority carriers at the junction approaches zero, and during t_f (the reverse-current fall time) the remainder of the unrecombined minority carriers injected while the diode was forward-biased are withdrawn. (The end of the period t_f is usually defined as the time when the reverse current reaches one-tenth of I_R.) The storage and fall times together are referred to as the diode recovery time. The storage and fall times are dependent upon the relative magnitudes of I_F and I_R, decreasing as I_F/I_R decreases.

The reverse recovery time of point-contact diodes is frequently of the order of nanoseconds. The shape of the recovery current wave is such as to make it difficult to distinguish the storage and fall times, and only the recovery time is normally measured. It has also been found that this recovery time is dependent upon the length of time that forward-bias current was applied. This effect may be a result of local junction heating.

The reverse recovery time may be defined as the time required for the reverse diode current to reach a given fraction of its initial value after the simultaneous removal of a given forward-bias current and the application of a given reverse-bias voltage through a given impedance. Another method of specifying reverse recovery time is in terms of the total extractable stored charge remaining in the diode after the removal of a forward bias.

A simple circuit which can be used to illustrate the principle of reverse-recovery-time measurement is shown in Fig. 10.32. It is assumed that the switch can be closed instantaneously, that precautions have been taken to reduce stray circuit capacitance and series inductance, and that the cathode-ray oscilloscope behaves as an infinite impedance-measuring device. When the switch is open a forward current flows in the diode given approximately by E_1/R_1, since the diode has low resistance and R_1 is at least an order of magnitude larger than R_2. Referring to the waveform of Fig. 10.32, at t_a the switch is closed and because of storage of minority carriers a reverse current of E_2/R_2 flows until the density of carriers at the junction reaches zero. At this time t_b the reverse current begins to fall exponentially toward its d-c value. The storage and fall times are indicated on the waveform.

A circuit to illustrate forward-recovery-time measurement is shown in Fig. 10.33. If a pulse of current is applied to the diode in the forward direction the initial voltage

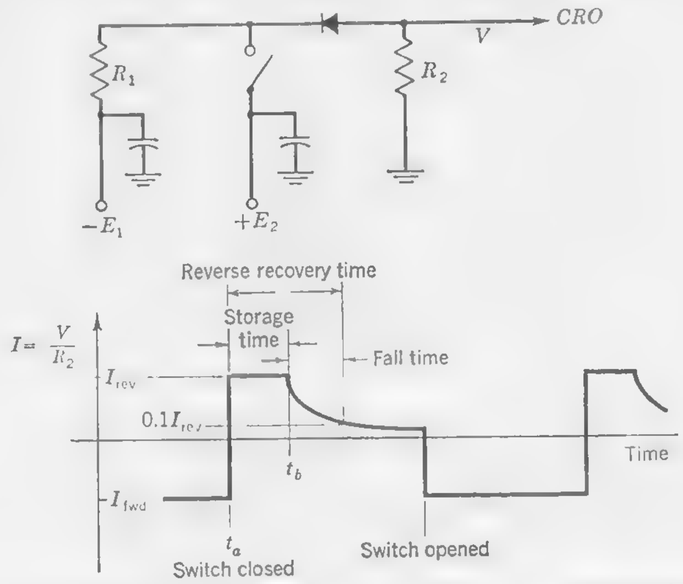


FIG. 10.32. Measurement of reverse recovery time, storage time, and fall time.

drop is high because of the semiconductor body resistance. This resistance decreases as injected carriers diffuse into the semiconductor regions adjacent to the junction resulting in conductivity modulation. The shape of the resulting diode-voltage-drop curve is practically independent of the prior reverse-bias condition of the diode (in contrast to storage time being greatly dependent on the previous forward current) except that a small but finite amount of current is required to discharge the capacitance of the reverse-biased diode.

The resistance R in Fig. 10.33 should be sufficiently large to ensure a constant current through the diode after the switch is closed. The voltage across the diode is then expected to have a waveshape as shown. A reasonable way to define forward recovery time is the time necessary for the voltage, after going through a maximum, to return to within 10 per cent of its final value. The peak forward voltage across the diode during the switching on transient is the "forward-peak-voltage transient."

10.8.9. Point-contact Diodes. Unlike transistors, semiconductor diodes have been known and studied for over eighty years. The existence of the semiconductor, as such, was discovered in 1931. As discussed in an excellent review article,[48] the first semiconductor diodes were contacts between metals and various sulfides such as galena and pyrites. Around 1900, it was found that point contacts on silicon gave the most stable diode detector of radio waves whereas galena detectors had the best sensitivity. World War II caused a great increase of interest in point-contact silicon diodes because of their usefulness as microwave detectors. These diodes were made

on very-low-resistivity silicon and hence could not stand high reverse voltage, which limited their usefulness in gating and clamping applications. The advent of the transistor spurred the research and development effort needed to produce single-crystal high-resistivity germanium, which was immediately adopted as a material with which to make high-reverse-voltage point-contact diodes for computer applications. Studies of forming followed which made possible point-contact diodes with reasonably good

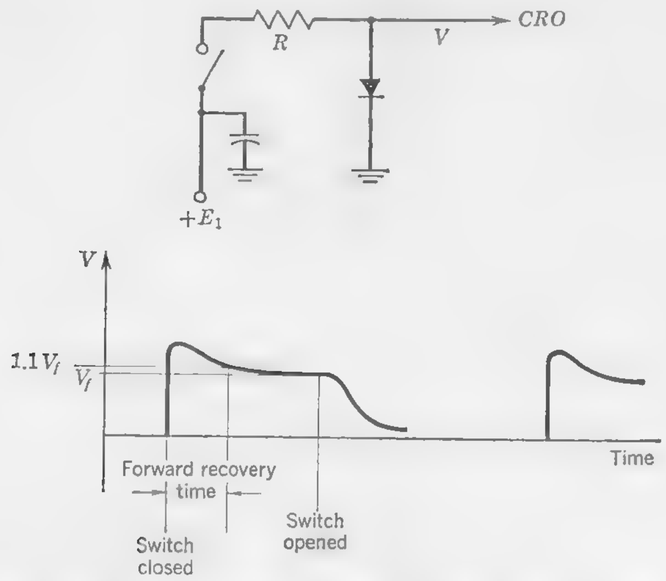


FIG. 10.33. Measurement of forward recovery characteristics.

forward and reverse characteristics. These devices still hold a prominent position as computer elements because of their low cost and inherent high speed. Their high speed results from a low capacitance as well as storage times in the millimicrosecond range.

10.8.10. Alloyed-junction Diodes.[27,55] As computer elements, small-area alloyed-junction diodes are of considerable interest. To be useful in gating circuits the capacitance and storage times must be low. Storage time decreases rapidly with the radius of the alloyed junction.[31] Gold-germanium alloy diodes are available in which the junction radius is a thousandth of an inch or less, which results in diodes with storage times as good as or better than point-contact germanium diodes. Larger-area germanium alloy diodes have been developed, primarily for rectifier application.

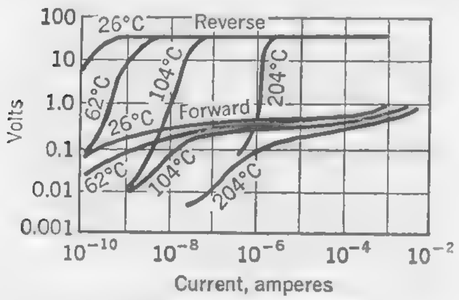


FIG. 10.34. Typical silicon diode characteristics, showing variation with temperature.

Aluminum-silicon alloyed-junction diodes[50] of several designs have been developed for low- and medium-speed switching applications and for use as voltage-limiting and clamping diodes. The capacitance is relatively high and varies inversely as the square root of the applied voltage. It is possible to design silicon alloy diodes with a wide range of breakdown voltages including the range of 3 to 200 volts. Probably the most important characteristic of alloy silicon diodes, however, is their extremely low reverse current at room temperature plus the fact that they can have a good rectification characteristic at temperatures as high as 200°C. Typical rectification characteristics[50] are shown in Fig. 10.34 for several values of ambient temperature. More detail on variation of breakdown voltage with temperature[50] is shown in Fig. 10.35.

10.8.11. Diffused-junction Diodes. Silicon diodes can be fabricated by diffusion of impurities into the solid semiconductor.[49] For example, a rectifying junction is obtained by the diffusion of phosphorus into p-type silicon. A good ohmic contact to the p-type silicon is obtained by diffusing boron into the opposite side of a thin silicon wafer. If the distance between the phosphorus diffused junction and the boron diffusion is sufficiently small, complete conductivity modulation is obtained under forward-bias conditions resulting in low forward voltage drop. Large-area diffused silicon diodes are being used as rectifiers and voltage regulators. Computer diodes capable of operating at high speed can be fabricated simply by limiting the area of the diffused junction and suitably reducing the silicon lifetime.

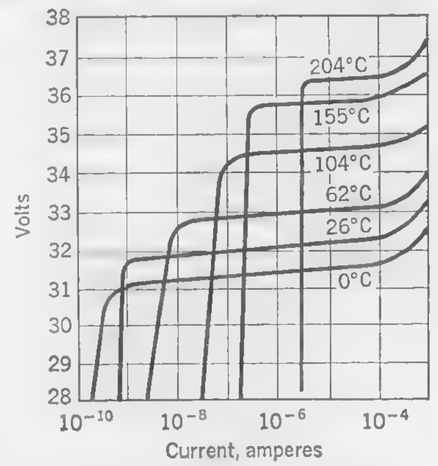


FIG. 10.35. Typical silicon diode breakdown characteristics, showing variation with temperature.

10.8.12. Typical Characteristics. Several thousand diode codes are available commercially. These include point-contact, alloy germanium, alloy silicon, and diffused silicon diodes. The electrical characteristics of typical switching diodes from each of these classes of devices are given in Table 10.8.

Table 10.8. Typical Limiting Characteristics of Various Diode Types

Type of diode	Max reverse sat. current at 10 volts, 25°C, μa	Max forward voltage at 10 ma, volts	Max forward current at 1.0 volts, ma	Max reverse recovery time with $I_F = I_R$ = 10 ma, μsec	Max capacitance at 5 volts, $\mu\mu$f	Min BV or PIV, volts
Point-contact Ge	10	2.0	5	0.05	2	80
Gold-bonded Ge	10	0.6	20	0.05	5	50
Alloy Ge	10	0.3	100	5	20	80
Diffused Ge	5	0.5	10	0.02	4	40
Alloy Si	0.02	0.8	30	0.05	10	70
Diffused Si	0.005	1.0	10	0.005	4	50
Diffused Si	0.02	0.6	250	0.05	25	120

The higher forward voltage drop indicated for silicon diodes as compared with similar germanium diodes results from an inherent property of silicon. (The energy gap of silicon is about 1.1 volts; the energy gap of germanium is about 0.7 volt.) Higher forward voltage drop is the price that has to be paid for lower reverse current and operability at higher temperatures.

In addition to the devices indicated in Table 10.8, silicon diodes are available with a wide range of breakdown voltages. The slope of the breakdown characteristic may be as low as 10 ohms for currents of the order of 10 ma.

10.9. TRANSISTORS*

Transistors can be divided into two major classes depending upon whether their short-circuit emitter-to-collector current gain α is less than or greater than unity.

* By J. J. Ebers and M. L. Embree.

10.9.1. Types of Transistors. Grown-junction, alloy-junction, surface-barrier, intrinsic-barrier, diffused-base, and field-effect transistors are in the $\alpha < 1$ class, while point-contact, hook-collector, double-base-diode, and avalanche transistors exhibit $\alpha > 1$.

Transistors with $\alpha < 1$ exhibit relatively stable d-c characteristics, low noise, and other characteristics which make them suitable for both analog and digital computers.

Transistors with $\alpha > 1$ exhibit a particularly desirable negative-impedance characteristic which makes possible two stable states with a single device. These transistors are useful in digital computers as regenerative pulse amplifiers, flip-flops, and other astable, bistable, or monostable circuits.

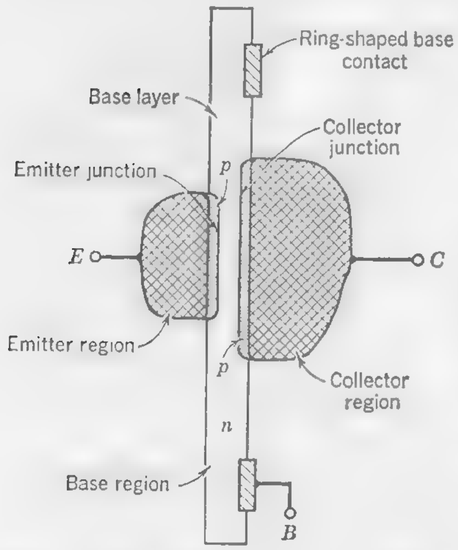


FIG. 10.36. Alloy-junction transistor structure.

10.9.2. General Description. Transistors are generally composed of three or more semiconductor regions.[62,65] These regions are characterized as n or p type depending upon the impurity concentration in the semiconductor. In an n-type semiconductor the current carriers are primarily electrons, while in a p-type semiconductor the majority carriers are holes.

In a p-n-p junction transistor the emitter and collector regions are p type. These p-type regions are separated by, and are contiguous to, the base region which is of the opposite conductivity type, that is, n type. Essentially ohmic contacts are made to the emitter, base, and collector regions. The boundary between the emitter region and the base region forms the emitter p-n junction, and the boundary between the collector region and the base region forms the collector p-n junction (see Fig. 10.36).

10.9.3. Symbolic Representation. Triode transistors are usually represented symbolically as shown in Fig. 10.37. The arrow on the emitter terminal represents the direction of conventional current flow when the emitter junction is forward-biased. The plus and minus signs represent the polarity of the voltage on the emitter and collector terminals relative to the base terminal for normal operation. The arrows at the extremities of the symbols indicate the assumed direction of positive conventional current flow.

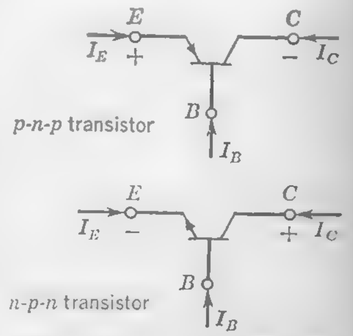


FIG. 10.37. p-n-p (upper) and n-p-n (lower) transistor symbols.

10.9.4. Reverse Cutoff Currents. Theoretically the current in a semiconductor p-n junction varies exponentially with applied voltage according to the relation[62]

$$I = I_S\left(e^{\frac{qV}{kT}} - 1\right) \tag{10.9.1}$$

where q is the charge on an electron, k is Boltzmann's constant, T is the temperature in degrees absolute (at 25°C, $kT/q = 0.026$ volt), V is the applied voltage, and I_S is a constant dependent upon the geometry, the constants of the material, and the temperature. With reverse-junction voltages* greater than $\frac{1}{10}$ volt, the exponential term becomes very small and a reverse current I_S, called the saturation current, flows. This current results when minority carriers in the regions adjacent to the junction diffuse over to the junction and are collected. In the case of conventional transistors it is convenient to define two reverse-bias cutoff n currents I_{CBO} and I_{EBO} for the collector and emitter junctions, respectively.[18,61] Since the two junctions are coupled through the base layer, it is necessary to define I_{CBO} as the reverse current of the collector junction with the emitter open-circuited and I_{EBO} as the reverse current of the emitter junction with the collector open-circuited.

* Reverse bias on a p-n junction implies p material negative with respect to n material.

In the case of $\alpha < 1$ transistors these currents are somewhat larger than the currents that would flow if both junctions were simultaneously reverse-biased, since the two junctions compete for the minority carriers in the base region.

10.9.5. Emission Efficiency γ. Under conditions of forward bias (p-type semiconductor positive with respect to n-type semiconductor) a p-n junction will emit minority carriers into both the adjacent regions.[62] The ratio of hole current to electron current will depend upon the conductivities of the two regions as well as the geometry of the transistor. In the case of the emitter junction it is desired that practically all the emitter current be minority carrier current injected into the base region. A factor γ (emission efficiency) is defined as the ratio of the minority carrier current to the total emitter current. Ideally γ should be unity.

10.9.6. Transport Factor β. Most of the minority carriers injected by the emitter diffuse across the base region into the space-charge region of the collector junction and are collected. A fraction, however, recombine within the base region or at the semiconductor surface. A factor β is defined as the fraction of the minority carriers emitted into the base region which are collected by the collector junction.[62]

10.9.7. Collection Factor α^*. Emitted carriers which diffuse across the base layer and find themselves in the collector space-charge layer drift across the region of high electric field and enter the collector body where they are majority carriers. For "alpha less than unity" transistors, the space-charge layer is thin, the carriers move with a high drift velocity, there is little chance for recombination, and the collection efficiency is high. There exist means by which collected carriers may be multiplied, for example, by avalanche multiplication, by a drift field in the collector body which draws minority carriers to the collector junction, or by a "hook-collector" junction. For this reason a collection factor α^* has been defined as the ratio of total collector current to incident minority carrier current. Thus the over-all current-transfer ratio a, emitter to collector, is given by

$$a = \gamma\beta\alpha^* \tag{10.9.2}$$

This is the factor which should appear in transistor equivalent circuits. It should be distinguished from the short-circuit current-transfer ratio α. Since these two quantities are so nearly equal for practical purposes in conventional transistors, no attempt will be made to separate them in the treatment to follow.

10.9.8. Frequency Cutoff of Alpha. Emitted carriers cross the base layer by a combination of diffusion and drift processes. Since a dispersion in transit time is inherent in both processes, changes in both magnitude and phase of alpha with frequency are to be expected. For a junction transistor it can be shown that alpha varies approximately as

$$\alpha = \frac{\alpha_0}{1 + j(f/f_\alpha)} \tag{10.9.3}$$

where α_0 is the low-frequency value of alpha and f_α is the frequency at which the magnitude of alpha falls to 1/2 of its low-frequency value.†

10.9.9. Junction Capacitance. Even under conditions of zero applied p-n junction bias an electrostatic potential exists at the junction in order to prevent carriers from diffusing in large numbers from the region where each type of carrier is in the majority to the region where it is in the minority. The electric field terminates on donor and acceptor ions resulting in a charge dipole at the junction.[62] The region of electric field is called the depletion layer since mobile charge has been swept out. There is a depletion-layer capacitance associated with the junction whose value is a function of the

† In some transistor designs the emitter efficiency may also affect the frequency behavior of the device. For point-contact transistors, the α usually falls off somewhat faster (at 7 to 9 db per octave) with frequency than Eq. (10.9.3) indicates.

applied voltage. As the applied junction voltage is varied, more or fewer donor and acceptor ions must be exposed in order to terminate the electric field, and hence the effective "spacing" of the capacitor varies.* In the case of step junctions (those in which the conductivity type changes abruptly), the depletion-layer capacitance varies inversely as the square root of the applied voltage. Alloyed-junction transistors usually have step junctions. In graded or gradual junctions, typical of grown-junction transistors, the depletion-layer capacitance varies inversely as the cube root of the applied voltage. Junctions obtained by an impurity-diffusion process usually lie somewhere in between.

In addition to the depletion-layer capacitance a junction diffusion or storage capacitance is associated with the minority carriers that are injected into the regions adjoining the forward-biased junction.[13] This capacitance can affect the high-frequency behavior of junction transistors.

The variation of the width of the depletion layer (alternatively called the space-charge layer) has a material effect on the transistor parameters. As the reverse bias on the collector junction is increased the collector junction effectively moves closer to the emitter which increases both alpha and its cutoff frequency.

10.9.10. Base Resistance. The base resistance r_b of a transistor is its reverse transfer or mutual resistance (see Fig. 10.43*a*).

In conventional transistors, some of the emitted carriers recombine in the base region and flow out the base contact as majority carrier current. Since the semiconductor base region has a definite resistivity, a resistance to this flow exists which is a major component of the base resistance.

In addition to the above component of base resistance, there is an emitter shielding effect of the collector field which adds to the base resistance. This effect is particularly important in point-contact transistors where the collector field can completely engulf the emitter at high collector voltages. This condition is somewhat analogous to "reach-through" (cf. Art. 10.9.19*d*) in alloy- and grown-junction transistors, and a rapid increase in base resistance is noted in these devices as the "reach-through" condition is approached.

At high frequencies, it is difficult to define a single base resistance. For $\alpha < 1$ transistors a network of resistances and capacitances has been derived to represent effects of base resistance at higher frequencies.

In general, for a given type of transistor, as the frequency response of the device is increased, the base resistance will also be increased.

10.9.11. Emitter and Collector Body Resistance. The emitter and collector regions have finite resistivities, particularly in grown-junction transistors, and it is necessary in some cases to specify emitter and collector body resistances.[18]

10.9.12. Avalanche Multiplication. The junction-rectification equation indicates that for reverse biases greater than about $\frac{1}{10}$ volt the current should remain constant. This is observed in junctions which are free of leakage currents. However, it is to be expected that at some voltage a mechanism should exist which would limit the reverse voltage which can be applied. Usually a multiplicative breakdown occurs which results from minority carriers colliding with the atoms of the crystal lattice in the high-field region of the junction to produce additional hole-electron pairs. The amount of multiplication depends upon the character of the junction and the applied voltage. At a given voltage for a particular junction the multiplication becomes infinitely large, and this voltage is therefore called the breakdown voltage. Since this is an avalanche or multiplicative process, it is referred to as avalanche breakdown.[40,42] For step junctions it has been shown that the multiplication can be empirically expressed as

$$M = \frac{1}{1 - (E/BV)^n} \tag{10.9.4}$$

* The variation of the width of the depletion layer (alternatively called the space-charge layer) has a material effect on the transistor parameters.[18] As the reverse bias on the collector junction is increased, the collector junction effectively moves closer to the emitter which increases both alpha and cutoff frequency.[14]

where E is the applied voltage, BV is the breakdown voltage, and n is a number dependent upon the characteristics of the junction.

Emitted carriers which diffuse across the base layer and into the collector-depletion layer are similarly multiplied.* The magnitude of this effect is dependent upon the transistor design. Frequently breakdown occurs near the surface and practically no multiplication of emitter current occurs.

10.9.13. Characterization. The characterization of an electronic device is the specification and determination of sufficient information to enable the prediction of the behavior of the device in a given class of applications. Usually transistors are characterized in a manner appropriate to the use for which they are designed, for example, for small-signal (linear) applications or for large-signal (nonlinear) applications. It is preferable and generally possible to characterize electronic devices in terms of parametric quantities such that the measurements depend only upon the internal parameters of the device.

Since transistor characterization necessarily depends directly on measuring techniques, additional characterization information on measurement and test methods will be in Arts. 10.9.17 to 10.9.21.

In switching or computing applications the transistor operates essentially as a switch.[18] The switch may normally be in the OFF condition (transistor in a low-current condition) and then switched to the ON condition (transistor in a high-current condition) by application of a pulse or step. Alternatively, the switch can be normally ON and then turned OFF by application of a pulse. A switch which changes state only during the application of an input pulse is said to be nonregenerative. Switches can of course be built which are regenerative in the sense that the application of a pulse "triggers" the switch to another state. In characterizing junction transistors for switching applications particular emphasis is placed on the nonregenerative type of operation, but the results are, in general, applicable to regenerative operation.

In the consideration of a transistor as a switch the important characteristics, as in all switches, are the OFF impedance, the ON impedance, trigger sensitivity, and the switching time. These switch characteristics are related to basic transistor parameters which are measurable at the transistor terminals and are independent of the circuit environment in which the transistor is placed.† To a good approximation the ON and OFF impedances can be obtained from the static characteristics or by making d-c measurements at several operating points. For $\alpha < 1$ transistors, the switch characteristics can also be related to the transistor reverse saturation currents and the forward and reverse alphas.‡ The switching, or pulse rise and fall times, can be related to the normal and inverse alphas and the normal and inverse alpha cutoff frequencies.[18,45,8] Collector capacitance, emitter impedance, and base impedance may also contribute to rise and fall times under some operating conditions.

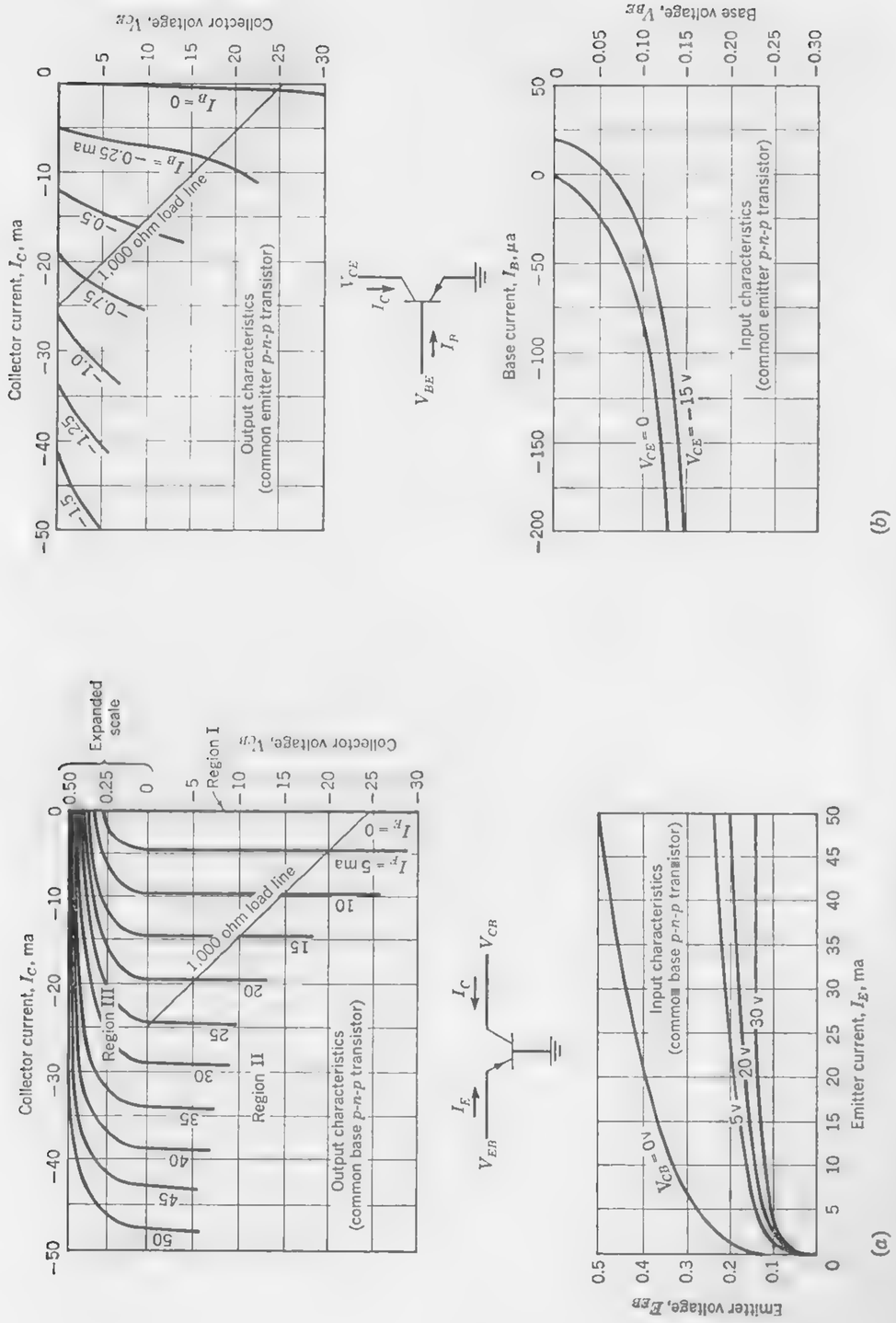
10.9.14. Large-signal Static Characterization. Transistors may be operated common (or grounded) base, common-emitter, or common-collector.[76] Transistors are sometimes characterized in the common-base connection, and from these data operation in the other modes can be derived. Common-base characterization has the advantage that the measured parameters are closely related to the physics of the device. The preponderance of switching circuits, however, use transistors in the common-emitter connection, and it is convenient to have available the common-emitter static characteristics as well as parameters relating to the transient behavior of the common-emitter transistor switch.

(*a*) *Static Characteristics.*[28] Electron tubes operated in the negative-grid region have essentially an infinite input impedance at low frequencies when operated common-

* This effect is the basic phenomenon of the avalanche transistor (cf. Art. 10.9.26).

† An exception is the case of operate times for transistors where characteristics, such as frequency response and capacity, are major functions of the instantaneous operating point. Under these circumstances, it may be found easier to make nonparametric tests of operate times in an appropriate circuit.

‡ When a transistor is manufactured, one terminal is marked the collector and another the emitter. For example, some transistors are symmetrical and the manufacturer can exercise choice in designating the collector. The inverse alpha α_i is the short-circuit current gain when the emitter is used as the collector and the collector is used as the emitter.



(b)

(a)

Fig. 10.38*a*. Common-base transistor static characteristics.
Fig. 10.38*b*. Common-emitter transistor static characteristics.

cathode. As a result, a single family of curves, usually the plate characteristics, is sufficient to determine completely the d-c or low-frequency behavior.

Transistors differ from vacuum tubes in this respect, and it is necessary to construct two families of curves. Since there are two currents and two voltages to be specified, four families of curves could be obtained but only two are necessary, and the sets which are normally chosen are the input and output characteristics. The most frequently used characteristics are

1. Common-base connection (see Fig. 10.38*a*)
   - *a*. Input characteristics obtained by holding the collector-to-base voltage constant and plotting emitter-to-base voltage against emitter current
   - *b*. Output characteristics obtained by holding the emitter current constant and plotting collector-to-base voltage against collector current
2. Common-emitter connection (see Fig. 10.38*b*)
   - *a*. Input characteristics obtained by holding the collector-to-emitter voltage constant and plotting base-to-emitter voltage against base current
   - *b*. Output characteristics obtained by holding the base current constant and plotting collector-to-emitter voltage against collector current

(*b*) *Regions of Operation.* There exists a large region of operation in the output characteristics in which the curves are essentially parallel and relatively equal in spacing. This region has been called the active region since a transistor biased in this region can exhibit gain when connected as an amplifier.

The active region, denoted as *region* II, is simply characterized by forward emitter-junction bias and reverse collector-junction bias. The collector current in the active region can be expressed as*

$$I_c = -\bar{\alpha}_f I_E + I_{CBO} \tag{10.9.5}$$

where $\bar{\alpha}_f$ is the large-signal or d-c value of the short-circuit forward current-transfer ratio of a transistor operated in the *normal* manner; that is, with emitter forward-biased and collector reverse-biased. The quantity $\bar{\alpha}_f$ usually shows significant variation with both collector voltage and emitter current. Since a junction transistor is composed of two junction diodes contiguous to the same base layer, it can also be operated in the *reverse* manner; that is, with the collector forward-biased and the emitter reverse-biased. Under these conditions it is possible to write

$$I_E = -\bar{\alpha}_i I_C + I_{EBO} \tag{10.9.6}$$

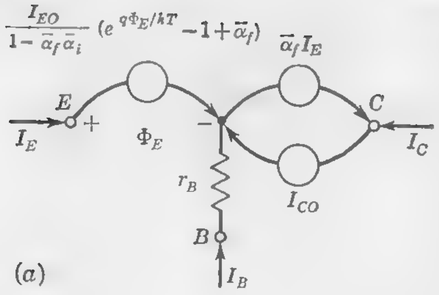
where $\bar{\alpha}_i$ is the large-signal value of the short-circuit forward current-transfer ratio of a transistor operated in the *inverse* manner.

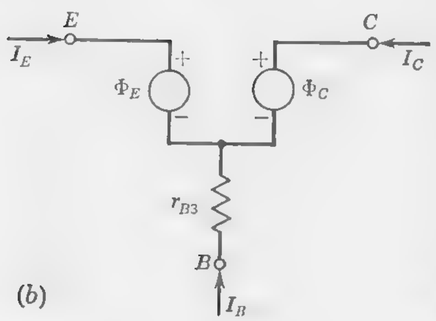
In switching or pulse applications the operating point may completely traverse the output characteristics along a load line such as that shown in Fig. 10.38*a*. Thus it is necessary to be able to specify the behavior of the transistor outside the active region. *Region* I has been defined as the region of the characteristics in which the transistor is in a low-current state, i.e., the switch is in the OFF condition (see Fig. 10.38*a*). This region has been called the current-cutoff region, and both emitter and collector junctions are reverse-biased.

(*c*) *Representation for D-C Behavior.* It has been shown that a single equivalent circuit can be used to predict the d-c behavior[18] of a junction transistor in regions I and II. The circuit, shown in Fig. 10.39*a*, will in fact trace out the common-base static characteristics to a good engineering approximation. For greater accuracy the variations of $\bar{\alpha}_f$ and $\bar{\alpha}_i$ must be considered as well as junction leakage currents and emitter and collector body resistances. In the circuit of Fig. 10.39*a*, Φ_E is the d-c voltage across the emitter junction.

At the other extreme of the load line the collector current is limited by the external circuit, resulting in a condition which has been termed collector-current saturation since an increase in emitter current does not further increase the collector current.

* Because of the assumed positive sense for I_E and I_C as shown in Fig. 10.37, I_{CBO} and I_{EBO} are positive quantities for n-p-n transistors and negative quantities for p-n-p transistors.





$$\Phi_E \approx \frac{kT}{q} \ln\left[-\frac{I_E + \bar{\alpha}_i I_C}{I_{EO}}\right]$$

$$\Phi_C \approx \frac{kT}{q} \ln\left[-\frac{I_C + \bar{\alpha}_f I_E}{I_{CO}}\right]$$

FIG. 10.39*a*. Transistor equivalent circuit for regions I and II, common base.
FIG. 10.39*b*. Transistor equivalent circuit for region III, common base.

Thus this region, *region* III, is characterized by forward bias on the emitter and external-circuit-limited collector current. For the case of $\alpha < 1$ transistors, both emitter and collector are forward-biased, since the collector junction must obtain a forward bias in order to reject the excess current injected into the base layer by the emitter.

An equivalent circuit for region III which holds for $\alpha < 1$ transistors is shown in Fig. 10.39*b*. Since the emitter and collector currents are the independent variables in region III, the junction voltages Φ_E and Φ_C appear as dependent variables. It is observed that different symbols are used for base resistance in the equivalent circuits of Fig. 10.39. This is because the base current flows in different paths and the region III base resistance may be significantly smaller than the region II base resistance.

The equivalent circuit shown in Fig. 10.40 will generate the common-emitter static characteristics for $\alpha < 1$ transistors shown in Fig. 10.38*b*. These are significantly different from the common-base characteristics in many respects. The curve corresponding to $I_B = 0$ corresponds to a variable forward bias on the emitter. The transistor is in the active region, and the collector current is approximately $I_{CO}/(1 - \alpha_f)$. In order to provide a switch with a low OFF current it is necessary to apply a bias on the base terminal in such a direction as to reverse-bias the emitter junction. The collector current may then be less than I_{CO}. In region III it is observed that the voltage drop from collector to emitter is very small. It is given by the equation

$$V_{CE(\text{sat})} = \pm \frac{kT}{q} \ln \frac{\bar{\alpha}_r[1 - (I_C/I_B)(1 - \bar{\alpha}_f)/\bar{\alpha}_f]}{1 + (I_C/I_B)(1 - \bar{\alpha}_r)} \qquad (10.9.7)$$

where the (+) sign applies to *p-n-p* transistors and the (−) sign to *n-p-n* transistors.

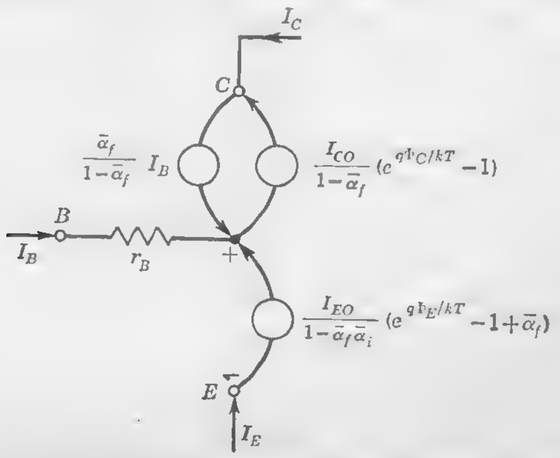
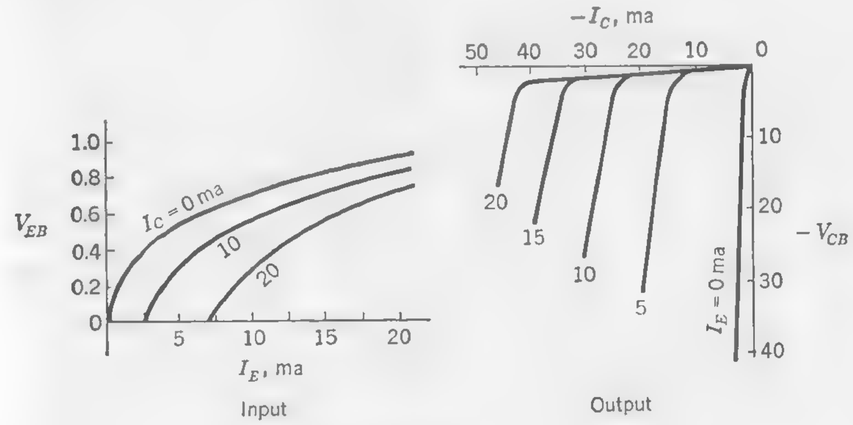


FIG. 10.40. Transistor equivalent circuit, common emitter.

Since $kT/q = 0.026$ volt at room temperature, the voltage drop across the switch in the common-emitter connection may be only a few millivolts for small values of I_C/I_B.

(*d*) *Negative-resistance Characteristics.* Transistors with $\alpha > 1$ generally exhibit negative-resistance characteristics in low-impedance circuits.[1,37] These negative-impedance characteristics are particularly useful not only for the design of circuits using this type of device but also for the characterization of the devices for the particular class of circuits in which they are commonly used. The emitter negative-resistance characteristics will be used as an example since it is the most common of the three.



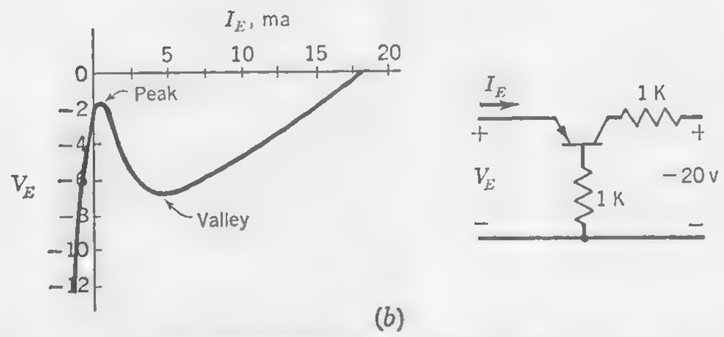


FIG. 10.41*a*. Common-base static characteristics (point-contact transistor).
FIG. 10.41*b*. Emitter negative-resistance characteristics (point-contact transistor).

Figure 10.41*b* shows a typical negative-resistance emitter characteristic, and the appropriate circuit. In this case external base and collector resistances of 1,000 ohms each have been selected. The operation of switching circuits utilizing the negative-resistance characteristics of $\alpha > 1$ transistors is such that the "trigger on" voltage is given by PV_E, and the "trigger off" or valley voltage is given by VV_E. The peak voltage PV_E is related to d-c transistor measurements and circuit values by the following formula:*

$$PV_E = V_{EB}(I_E = 50\ \mu\text{a},\ V_C) + I_C(I_E = 0,\ V_C)R_B \qquad (10.9.8)$$

By making certain simplifying assumptions the valley voltage VV_E can be related to the circuit values and the large-signal alpha of the transistor. Those assumptions are as follows:

$$R_B \gg r_{B1,2,3} \text{ or } r_{E2,3} \text{ or } r_{C3} \quad \text{and} \quad R_B \ll r_{C2} \text{ or } r_{E1}$$

* $I_C(I_E, V_C)$ means collector current I_C measured at an emitter current I_E and a collector voltage V_C.

where* R_B = external base resistance

$r_{B1,2,3}$ = internal base resistance in regions 1,2,3, respectively
$r_{E1,2,3}$ = internal emitter resistance in regions 1,2,3, respectively
$r_{C2,3}$ = internal collector resistance in regions 2,3, respectively

With the above assumptions, the emitter voltage is given by the following relationships:

Region I:

$$V_{E1} \cong PV_E - r_{E1}I_E \tag{10.9.9}$$

where

$$r_{E1} = \frac{V_E}{I_E(V_E,\ I_C = 0)} \tag{10.9.10}$$

Region II:

$$V_{E2} \cong PV_E + R_B(1 - \bar{\alpha}_{fb})I_E \tag{10.9.11}$$

where $\bar{\alpha}_{fb}$ = large-signal common-base forward short-circuit transistor current gain

Region III:

$$V_{E3} \cong \frac{V_{CC}R_B}{R_B + R_C} + \frac{R_BR_C}{R_B + R_C} I_E \tag{10.9.12}$$

where R_C = external collector load resistance
V_{CC} = collector supply voltage

Solving the last two of the above equations simultaneously, the valley voltage (turn-off voltage) point VV_E is found to be

$$VV_E \cong \frac{PV_ER_C - V_{CC}R_B(1 - \bar{\alpha}_{fb})}{R_C - (1 - \bar{\alpha}_{fb})(R_B + R_C)} \tag{10.9.13}$$

A more general treatment of negative-resistance characteristics can be found in the literature.

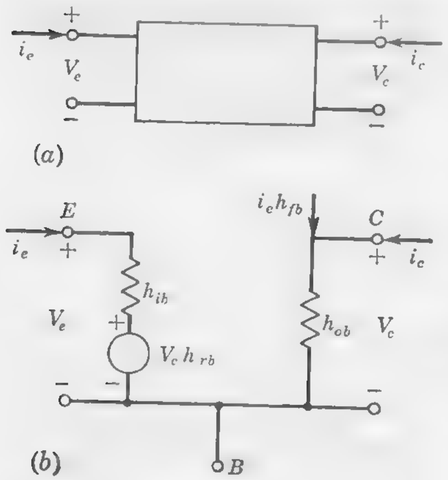


FIG. 10.42*a*. Transistor as a "black box."
FIG. 10.42*b*. Transistor small-signal equivalent circuit.

10.9.15. Small-signal Transient Characterization. The transient behavior of transistors can be approximated in terms of small-signal parameters. It is possible to obtain better accuracy in some cases using large-signal pulse measurements, but on the other hand, the only linear analysis possible is in terms of small-signal parameters. In order to make a linear analysis, it is assumed that these parameters remain constant within any one of the three regions of operation.

Historically, the first small-signal parameters to be measured were open-circuit impedance measurements whose low-frequency values are the slopes of the two families of curves shown in Fig. 10.38*a* and the other two families of this set which are related to the transfer impedances.[65,76,1,56] The small-signal parameters which are now in common use for $\alpha < 1$ transistors are the hybrid, or h, parameters defined by the following equations for the transistor considered as a black box as shown in Fig. 10.42*a*:[28]

$$\begin{aligned} v_e &= h_{ib}i_e + h_{rb}v_c \\ i_c &= h_{fb}i_e + h_{ob}v_c \end{aligned} \tag{10.9.14}$$

* For additional explanation of these small-signal parameters see Art. 10.9.15 and References 76, 28, 1, 37, 56, 22, and 51.

The second subscript refers to the mode of operation, in this case common base. The h parameters are thus defined as

$h_{ib} = \frac{v_e}{i_e}$	when $v_c = 0$	INPUT IMPEDANCE WITH OUTPUT TERMINALS A-C SHORT-CIRCUITED
$h_{rb} = \frac{v_e}{v_c}$	when $i_e = 0$	VOLTAGE FEEDBACK RATIO WITH INPUT TERMINALS A-C OPEN-CIRCUITED
$h_{fb} = \frac{i_c}{i_e}$	when $v_c = 0$	CURRENT FORWARD-TRANSFER RATIO WITH OUTPUT TERMINALS A-C SHORT-CIRCUITED
$h_{ob} = \frac{i_c}{v_c}$	when $i_e = 0$	OUTPUT ADMITTANCE WITH INPUT TERMINALS A-C OPEN-CIRCUITED

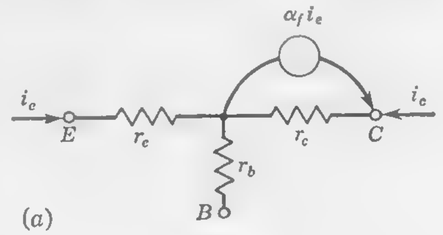
The above equations lead to the equivalent circuit shown in Fig. 10.42*b*. At low frequencies this is equivalent to the circuit shown in Fig. 10.43*a*, the r's and h's being related by the equations

$$h_{ib} = r_e + (1 - \alpha)r_b \tag{10.9.15}$$

$$h_{rb} = \frac{r_b}{r_c} \tag{10.9.16}$$

$$h_{fb} = -\alpha \tag{10.9.17}$$

$$h_{ob} = \frac{1}{r_c + r_b} \approx \frac{1}{r_c} \tag{10.9.18}$$



For computational convenience, the value of r_e in Eq. (10.9.15) may be calculated from the relation

$$r_e = \frac{kI}{qI_E}\frac{1}{1 + \alpha} \tag{10.9.19}$$

which neglects any emitter body resistance which may exist. Thus r_e is about 13 ohms for an emitter-bias current of 1 ma at room temperature (25°C).

The circuit of Fig. 10.43*a*, modified as in Fig. 10.43*b* to take into account the frequency behavior of alpha, collector capacitance, and the frequency behavior of the emitter impedance, has been used to derive expressions for pulse rise and fall times for $\alpha < 1$ transistors.[45] Since Y_e is a large admittance, its effect on the output waveform will be negligible if the transistor is driven by a high-impedance source. However, it must be considered if it is desired to relate the input current and voltage. The quantity $C_{j\omega e}$ is the emitter storage or diffusion capacitance.

FIG. 10.43*a*. Low-frequency transistor equivalent circuit.
FIG. 10.43*b*. Transistor equivalent circuit for pulse-rise-time analysis.

10.9.16. Large-signal Transient Characterization.[18,45,23,69] One of the most often used transistor pulse-amplifier circuits is shown in Fig. 10.44. This is simply a common-emitter pulse amplifier with a resistive load and driven by a high-impedance source. In order to obtain a short pulse delay and rise time, the transistor is driven from current cutoff completely into current saturation. A large number of minority carriers are injected into the base region* (and also into the collector region for grown-junction and diffused-base transistors). The actual density of minority carriers in the base region may be a factor of ten or a hundred times its value when the transistor is in the active region. If the base current is now returned to zero or reversed, the

* This situation is aggravated in the case of $\alpha < 1$ transistors, since the collector as well as emitter junction is forward-biased and therefore emits minority carriers into the base region.

density of injected carriers in the base region near the collector junction must be reduced to zero before the collector junction can become reverse-biased. The current in the load remains essentially constant for a period of time called the storage time, since it is attributable to the storage of minority carriers in the base region. When the density of carriers near the collector junction approaches zero, the load current begins to decrease. The pulse fall or decay time is determined by the frequency characteristics of the transistor and the magnitude of the base drive. The chain of events is shown in Fig. 10.44. Both base and collector currents are shown on the same time scale.

In characterizing the pulse behavior of transistors, it is often advantageous simply to measure the operating times (t_d, t_r, t_s, and t_f) under conditions similar to those encountered in the particular class of circuits for which the transistor is characterized.

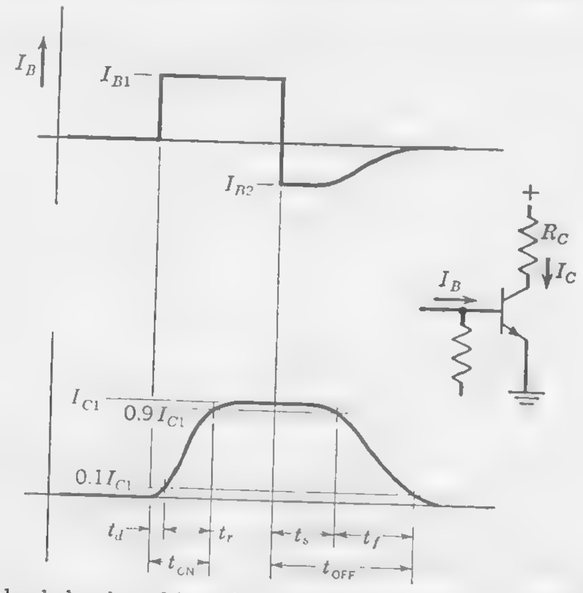


FIG. 10.44. Pulse behavior of junction transistor (common-emitter connection).

These measurements are generally nonparametric and are difficult to apply to circuits other than those of the particular class for which they are designed.

The delay, rise, storage, and fall switching times are critically dependent upon the mode of operation of the transistor, the circuit, the drive and upon the magnitude of the biases. Significant differences in transient response arise depending upon whether the device is driven by a constant current or constant voltage source.

An equivalent circuit that has been used to calculate rise and fall times for $\alpha < 1$ transistors in the common-emitter connection is shown in Fig. 10.43*b*. Using this circuit and others suitable for predicting behavior in regions I and III, delay, rise, storage, and fall times have been calculated[45,47,67] for a common-emitter amplifier with a resistive load driven from a high-impedance source. These results, algebraically modified for clarity, are

$$t_d = \frac{C_C + C_E}{I_{B1}} \left| V_{BEO} - V_{BE1} \right| + \tau_r \ln \frac{1}{1 - 0.1\, I_{BS}/I_{B1}} \tag{10.9.20}$$

$$t_r = \tau_r \ln \frac{1 - 0.1\, I_{BS}/I_{B1}}{1 - 0.9\, I_{BS}/I_{B1}} \tag{10.9.21}$$

$$t_s = \tau_s \ln \frac{1 - I_{B1}/I_{B2}}{1 - I_{BS}/I_{B2}} + \tau_r \ln \frac{1 - I_{BS}/I_{B2}}{1 - 0.9\, I_{BS}/I_{B2}} \tag{10.9.22}$$

$$t_f = \tau_r \ln \frac{1 - 0.9\, I_{BS}/I_{B2}}{1 - 0.1\, I_{BS}/I_{B2}} \tag{10.9.23}$$

where

$$\tau_r = \frac{1 + 2\pi f_{\alpha f}\, R_C\, C_C}{2\pi f_{\alpha f}(1 - \bar{\alpha}_f)} = \text{rise time constant} \tag{10.9.24}$$

$$\tau_s = \frac{f_{\alpha f} + f_{\alpha i}}{2\pi f_{\alpha f} f_{\alpha i}(1 - \bar{\alpha}_f \bar{\alpha}_i)} = \text{storage time contant} \tag{10.9.25}$$

$$I_{BS} = \frac{I_{C1}}{\bar{\alpha}_f/(1 - \bar{\alpha}_f)} = \text{minimum saturation base current} \tag{10.9.26}$$

The currents have the significance shown in Fig. 10.44. The quantities $f_{\alpha f}$ and $f_{\alpha i}$ are the frequency cutoffs of the short-circuit current gain in the normal and inverse connections. C_C and C_E are the collector and emitter depletion capacitances, respectively. These equations may be used to predict the switching times with an accuracy generally sufficient for engineering analysis.

10.9.17. Methods of Measurement and Test. Standardization of methods of test is a problem which has been considered by governmental agencies, both civilian and military, as well as by industrial and professional organizations. Proposed standards have appeared in the literature.[28] Because of the complexity of the entire problem, a division has been made between small-signal and large-signal tests. As stated previously, transistors may be designed particularly for either large-signal or small-signal applications, which further justifies the separation.

10.9.18. Large-signal Measurements. Large-signal measurements are particularly valuable in the characterization of transistors for switching applications such as digital-computer circuitry. The large-signal measurements can be conveniently divided into three classes: static characteristic curves, d-c measurements, and transient measurements.

Static characteristic curves are particularly valuable since they provide a picture of the behavior of the transistors over wide ranges of operating conditions.

(*a*) *Point-by-point Measurements of Static Characteristic Curves.* The static characteristic curves of a transistor in the common-base connection may be obtained using the circuit shown in Fig. 10.45*a*.* One of the parameters, e.g., I_E, is held constant while another, possibly V_C, is varied, and readings are taken of the other two parameters.

(*b*) *Swept Static Characteristic Curves.* In many cases it is advantageous to display the characteristic curves on an oscilloscope. The circuit shown in Fig. 10.45*b* can be used to view the collector characteristic curves for a given value of emitter current. The sinusoidal supply varies the collector voltage, and a collector current vs. collector voltage trace is obtained on the oscilloscope.

In switching or pulse applications, the collector characteristics may be of interest in areas of the active region where they cannot be obtained by d-c or sinusoidal means since the power-dissipation ability of the transistor may be exceeded. In this case pulse data must be obtained.

(*c*) *Negative-resistance Static Characteristic Curves.* Negative-resistance characteristic curves (see Fig. 10.41*b*) for $\alpha > 1$ transistors can be measured by using the same methods indicated above except that a circuit including a transistor is measured instead of a transistor alone. Precautions outlined in the preceding footnote also apply for these measurements.

10.9.19. D-C Measurements. D-C "point-type measurements"† are quite convenient and relatively simple. They are particularly desirable for use in the production and specification of transistors in order to control the necessary characteristics to assure proper operation in the class of circuit for which the transistor is designed. The relative simplicity of this type of measurement also makes them desirable from the standpoint of the circuit designer since correlations between device characteristics and circuit operations are simplified.

* When measuring the characteristics of $\alpha > 1$ transistors, regions of instability and/or oscillation may be encountered which result from the short-circuit or open-circuit instability characteristics of such transistors. Generally the use of low-shunt-capacity noninductive resistors in series with each of the leads at the emitter and a small shunting capacitance between collector and base will prevent instability.

† "Point-type measurements" are those taken at a particular operating point.

(*a*) *Current Gain.* The large-signal forward common-base current gain $\bar{\alpha}_{fb}$ and the large-signal forward common-emitter current gain $\bar{\alpha}_{fe}$ are defined by the following relations

$$\bar{\alpha}_{fb} = \frac{I_C(I_E,V_{CB}) - I_C(0,V_{CB})}{-I_E} \tag{10.9.27}$$

$$\bar{\alpha}_{fe} = \frac{I_C(I_B,V_{CE}) - I_C(0,V_{CE})}{I_B} \tag{10.9.28}$$

The static value of the forward common-base current gain h_{FB} and the static value of the forward common-emitter current gain h_{FE} are defined by Eqs. (10.9.27) and (10.9.28) above if $I_C(0,V_{CB})$ and $I_C(0,V_{CE})$ are respectively equal to zero. As indicated in the above formulas, the only measurements required in order to evaluate large-signal and static-value current gains are the collector current for zero emitter or base current, as the case may be; the collector current for the desired emitter or base current; and, of course, the emitter or base current. In the case of $\alpha < 1$ transistors, the large-signal inverse-current gain is also of importance. Both the common-base and common-emitter inverse-current gains ($\bar{\alpha}_{ib}$ and $\bar{\alpha}_{ie}$) can be obtained by utilizing formulas similar to those given above except that the functions of collector and emitter are interchanged.

The circuit of Fig. 10.45*c* can be used to make both the forward and the inverse static value and approximate large-signal current-gain measurements for $\alpha < 1$ transistors. At room temperature or below, I_{CO} and I_{EO} are usually sufficiently low in comparison with the input current that they may be neglected. To a good approximation the meter (I_B) will read $(1 - \bar{\alpha}_{fb})I_E$ or $(1 - \bar{\alpha}_{ib})I_C$, and the large-signal current gains are approximately equal to the static-value gains.

For $\alpha > 1$ transistors, a convenient approximation to the large-signal common-base forward current gain $\bar{\alpha}_{fb}$ can be obtained by measuring the collector current at a relatively low collector voltage and a high emitter current. Under these conditions, the circuit of Fig. 10.45*a* may be used and the effect of I_{CBO} may be neglected and $\bar{\alpha}_{fb} \approx h_{FB}$.

(*b*) *Cutoff Currents* (OFF *Currents*). Two of the most important d-c parameters are I_{EBO} and I_{CBO}. They can be measured by applying a voltage emitter to base, or collector to base with polarity appropriate for reverse bias, and recording the resulting current. It is usually desirable to provide some method of limiting the current from the constant-voltage source in case of faulty transistors. Usually I_{EO} and I_{CO} are measured at a voltage about one-half the maximum recommended emitter and collector voltages, respectively. The complete voltage-current curve of the collector and emitter may also be traced out on an oscilloscope. For many transistor types, cutoff currents are measured at a temperature comparable with the maximum temperature the transistor is expected to experience in operation.

(*c*) *Breakdown Voltage.* As the reverse bias on a junction is increased, a voltage is eventually reached at which the reverse current increases very rapidly.[40,42,89] In well-behaved junctions the breakdown is relatively sharp and is due to avalanche breakdown in the body of the semiconductor or in the vicinity of the surface. Breakdown voltage is normally specified as the reverse junction voltage for a given current, such as 50 or 100 μa. The test can be instrumented by biasing the junction from a high-impedance constant-current source or by tracing out the reverse characteristics on an oscilloscope and noting the voltage at which the current attains the specified value. The breakdown voltages have been denoted as BV_{CBO} and BV_{EBO} for the collector-to-base and emitter-to-base breakdown voltages, with open emitter and collector, respectively.

In point-contact transistors, breakdown voltages are usually not observed, since the maximum power dissipation of the device is usually exceeded before the breakdown voltage can be reached.

(*d*) *Reach-through Voltage* (*Punch-through Voltage*). In many transistors the space-charge-layer widening of the collector (or emitter) junction, under conditions of reverse bias, is significant; and the space-charge region extends across the base layer to the

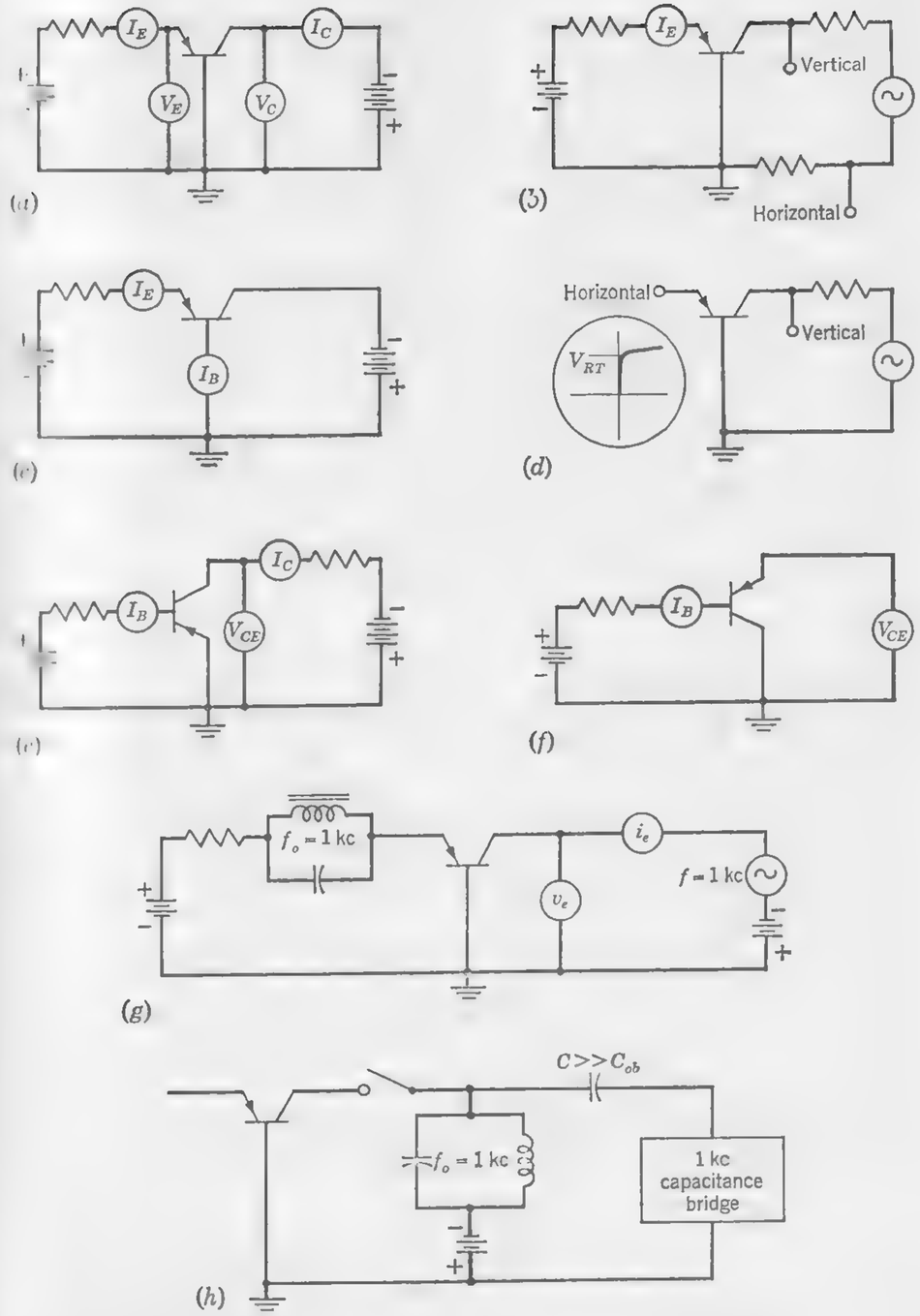


Fig. 10.45*a*. Circuit for measurement of static characteristics.
Fig. 10.45*b*. Circuit for viewing output characteristics on an oscilloscope.
Fig. 10.45*c*. Circuit for measurement of static value of current gain ($\bar{\alpha}_{fb} \approx h_{FB}$ and $\bar{\alpha}_{fe} \approx h_{FE}$).
Fig. 10.45*d*. Circuit for measurement of reach-through voltage.
Fig. 10.45*e*. Circuit for measurement of collector-to-emitter saturation voltage.
Fig. 10.45*f*. Circuit for measurement of collector body resistance.
Fig. 10.45*g*. Circuit for measurement of small-signal output impedance h_{ob}.
Fig. 10.45*h*. Circuit for measurement of output capacitance C_{ob}.

emitter (or collector) junction.[63] Under these conditions, a low resistance is observed between the emitter and collector. This phenomenom has been called electrical reach-through (punch-through). In most transistors the reach-through voltage is greater than the breakdown voltage and hence is not observed. Reach-through voltage may be measured using the circuit of Fig. 10.45*d*. D-C meters may be used to determine the reach-through voltage (if it exists); however, the scope method is particularly desirable under certain circumstances. The same value for V_{RT} (reach-through voltage) is generally obtained in the case of $\alpha < 1$ transistors whether emitter or collector junction is reverse-biased.

Another convenient method of obtaining an appropriate measure of the reach-through voltage is to measure the emitter breakdown voltage of the transistor with collector and base shorted together. In this case the measured value is either the emitter breakdown or reach-through voltage, whichever is smaller. The reach-through voltage is a major limitation in the case of high-frequency alloy-junction transistors because of the thin base layer.

(*e*) *Sustaining Voltage.* Because of avalanche multiplication $\bar{\alpha}_{fb}$ and/or $\bar{\alpha}_{ib}$ may become greater than unity for voltages below the breakdown voltage.[17,43] The voltage at which $\bar{\alpha}_{fb}$ becomes unity has been called the sustaining voltage $V_{CE(sus)}$. This voltage can be measured by observing the collector-to-emitter voltage (with the base open) at which the current increases without limit since

$$I_C = \frac{MI_{CO}}{1 - \bar{\alpha}_{fb0}M} \tag{10.9.29}$$

Under these conditions and as soon as the multiplication M times the low-voltage current gain $\bar{\alpha}_{fb0}$ becomes unity, I_C increases without limit. The voltage $V_{CE(sus)}$ may represent a maximum limit on the operating voltage in some applications. A convenient method of measuring the sustaining voltage $V_{CE(sus)}$ is merely to measure the common-emitter collector breakdown voltage with an open base.

(*f*) *Saturation Voltage* (ON *Voltage*). For $\alpha < 1$ transistors, Eq. (10.9.7) may not give a very accurate measure of the collector-to-emitter saturation voltage $V_{CE(sat)}$ at high current levels since $\bar{\alpha}_{fb}$ and $\bar{\alpha}_{ib}$ vary appreciably with current.[18,17] For this reason it may be necessary to measure V_{CE} in region III at current levels approximating those which may be encountered in a specific application. This can be done by using the circuit of Fig. 10.45*e*.

For $\alpha > 1$ transistors, both the region III collector-to-emitter saturation voltage $V_{CE(sat)}$ and the collector-to-base saturation voltage $V_{CB(sat)}$ are found to be useful. Conventional d-c measurement techniques can be used to evaluate these parameters.

(*g*) *Region III Base Resistance.* In many pulse applications the d-c base resistance in region III, r_{B3}, is of particular importance.[18] Neglecting body and/or spreading resistance that may exist in the collector and emitter regions, r_{B3} may be calculated from a measurement of the forward emitter-to-base characteristic. The value r_{B3} may be obtained by measuring the voltage drop at a given current and subtracting out the junction voltage drop, which may be obtained from the equation

$$V_j = \frac{kT}{q} \ln \frac{I_E + I_{EO}}{I_{EO}} \tag{10.9.30}$$

and calculating the desired resistance by the equation

$$r_{B3} = \frac{V_{BE} - V_j}{I_B} \tag{10.9.31}$$

If only a qualitative result is required, the emitter-to-base voltage drop for a given current may be used directly as a control of the region III base resistance.

(*h*) *Body Resistance.* Some junction transistors, particularly grown-junction transistors, may have appreciable collector or emitter body resistance.[18] The collector body resistance r_{Cv} (or r_{Ev}) may be measured using the circuit of Fig. 10.45*f*. When the transistor is operated in this manner (open-collector), the emitter obtains a forward

bias almost equal to the collector forward bias, and the collector body resistance is given by

$$r_{Cv} = \frac{V_{CE}}{I_B} \tag{10.9.32}$$

10.9.20. Transient Measurements. In Fig. 10.44 a common transistor pulse circuit is shown. In addition, the waveforms obtained with this pulse circuit are indicated. On this waveform the four components of operating time are indicated. These components are the delay time t_d, the rise time t_r, the storage time t_s, and the fall time t_f.*

In the case of $\alpha < 1$ transistors, analytical expressions are available to relate the operating times of the transistors to the large-signal alphas, their frequency responses, and various circuit currents. However, in the case of $\alpha > 1$ transistors such analytical relations are not available, and it has been found expedient to make direct measurements of the turn-on, or delay plus rise time, and the turn-off, or total of storage and fall times. The direct-measurement method is also commonly used for the characterization of $\alpha < 1$ switching transistors which are intended for use in a particular class of switching circuits.

(*a*) *Transient Measurements for* $\alpha < 1$ *Transistors.* As indicated by Eqs. (10.9.20) to (10.9.23), the pulse switching times are intimately related to $\bar{\alpha}_f$, $\bar{\alpha}_i$, $f_{\alpha f}$, $f_{\alpha i}$ for $\alpha < 1$ transistors. The d-c values $\bar{\alpha}_f$ and $\bar{\alpha}_i$ may be used to calculate switching time with fairly good accuracy.[45] The parameters $f_{\alpha f}$ and $f_{\alpha i}$, measured by small-signal methods, may be used in the calculations. However, all these parameters vary with operating point, and since the operating point traverses such a large range of the characteristics in pulse circuits, it is frequently preferable to obtain measurements of these important parameters by pulse methods.

The four switching times can be directly measured using circuits similar to that shown in Fig. 10.44. For the measurement of delay time (t_d), the base-to-emitter voltage prior to turn-on (V_{BEO}), the collector supply voltage (V_{CC}), the collector load resistance (R_C), and the turn-on base-drive current (I_{B1}) must be given. For rise time (t_r), only V_{CC}, R_C, and I_{B1} are required. For storage time (t_s), V_{CC}, R_C, I_{B1}, and the turn-off base current I_{B2} are needed. Finally, for the measurement of fall time (t_f), V_{CC}, R_C, I_{B1}, and I_{B2} are needed in order to establish the measurement conditions.

Switching transistors can be characterized in terms of the rise and storage time constants τ_r and τ_s. These time constants can be measured by measuring t_r or t_f and t_s and calculating the desired time constants using Eq. (10.9.21) or (10.9.23) and (10.9.22). When transistors are characterized using these time constants, the collector load resistance R_C should be chosen such that $2\pi f_{\alpha f} R_C C_C \ll 1$ in order to make the rise time constant τ_r as parametric as possible.

(*b*) *Transient Measurements for* $\alpha > 1$ *Transistors.*[19] The turn-on time (sum of delay and rise times) and the turn-off time (sum of the storage and fall times) of $\alpha > 1$ transistors can be measured using a grounded emitter circuit such as that shown in Fig. 10.44. A square-wave generator is applied through a capacitor and a resistor in series to the base of the transistor. The square wave periodically turns the transistor ON and OFF. The turn-on and turn-off times are then measured directly by using a triggered oscilloscope to observe the collector waveform. The base-current drive, collector load resistance, and collector supply voltage are specified in accordance with the class of circuits for which the transistor is designed.

10.9.21. Small-signal Measurements.[28] The low-frequency† values of small-signal parameters may be obtained by measuring the slopes of the static characteristics.

* Definitions of delay time, rise time, storage time, and fall time are as follows: Delay time t_d is the time between the start of the input pulse and the time when the output pulse has increased to 10 per cent of its maximum amplitude; rise time t_r is the time duration during which the output pulse is increasing from 10 to 90 per cent of its maximum amplitude; storage time t_s is the time between the end of the input pulse and the time when the output pulse has decreased to 90 per cent of its maximum amplitude; and fall time t_f is the time duration during which the output pulse is decreasing from 90 to 10 per cent of its maximum amplitude.

† 1,000 cps is the standard for low-frequency measurements.

Usually, however, standard a-c methods are used because of the greater accuracy obtained and the possibility of direct measurement.

(*a*) *Hybrid Parameters.* The a-c measurement of small-signal parameters is performed as dictated by the parametric equations which describe the four-terminal network* [see Eq. (10.9.13)]. For example,

$$h_{ob} = \frac{i_c}{v_c} \qquad \text{when } i_e = 0 \tag{10.9.33}$$

To perform the measurement, bias is applied to the transistor to establish the operating point. Then a small a-c signal is applied to the output terminals and the resulting output voltage and current are measured. Care must be taken that the emitter circuit is open-circuited with respect to the applied a-c signal. The circuit is shown in Fig. 10.45*g*. By varying the amplitude of the applied signal and observing the ratio of i_c/v_c it is possible to determine the linearity of the device at the chosen operating point and whether or not a true small-signal measurement is being obtained. The frequency of the a-c signal can obviously be varied to determine the frequency behavior of the parameter being measured. The phase angle of the parameter in question can be measured by established techniques.

(*b*) *Capacitance.* The collector capacitance may be measured using the circuit of Fig. 10.45*h*. The capacitance C_{ob} is the difference between the capacitance bridge reading with the transistor in, and with the transistor out of the circuit. 1 kc, 100 kc, and 1 mc are commonly used test frequencies.

10.9.22. Device Ratings and Rating Systems. Ratings are limiting environmental conditions (thermal, electrical, mechanical, etc.) which, if exceeded, may result in permanent impairment of device performance. Each rating is independent of other ratings; i.e., any condition which results in the exceeding of any one rating may result in permanent damage to the device. In general, ratings are established by extensive testing by the device manufacturer for the purpose of protecting the device from misuse due to environmental conditions under the control of the user.

The absolute-maximum rating system† is particularly suitable for those semiconductor parameters for which rapid degradation occurs when a particular environmental condition is exceeded; e.g., when the breakdown voltage of a transistor is exceeded, immediate failure may result. In addition, with this system, the device manufacturer can determine the necessary limit values by life and environmental testing of the device without considering the circuit in which the device will eventually be used.

When only absolute-maximum ratings are supplied by the device manufacturer, the circuit designer must design his circuit such that none of the absolute-maximum value is exceeded under any possible circumstances.

An additional system, recommended-maximum ratings,‡ has been devised to cover those cases where it is unusual to reach the absolute-maximum value for a particular parameter because the absolute-maximum value for another condition is normally reached first; e.g., as the collector voltage of a point-contact transistor is increased, the absolute-maximum collector power dissipation will normally be reached before an

* When small-signal measurements of $\alpha > 1$ transistors are required, the open-circuit impedances z_{ib}, z_{rb}, z_{fb}, and z_{ob} may be preferred, since this type of transistor may exhibit instability when attempting short-circuit measurements such as h_{ib} and h_{fb}.

† Absolute-maximum ratings may be defined as the limiting values beyond which the serviceability of the semiconductor device may be impaired from the viewpoint of life and satisfactory performance. The equipment designer must establish the circuit design so that initially and throughout the semiconductor device and equipment life, no absolute-maximum value is exceeded under the worst probable operating conditions with respect to supply-voltage variation, equipment-component variation, equipment-control adjustment, load variation, signal variation, environmental conditions, and variations in semiconductor-device characteristics.

‡ A recommended-maximum rating for a particular parameter may be defined as the limiting value beyond which the absolute-maximum value for some other parameter would be expected to be exceeded when normal-device characteristic variations and customary circuits are assumed.

excessive collector voltage can be reached. The recommended-maximum values are intended to prevent the circuit designer from being misled by the quotation of unattainable maximum values. They are normally used to supplement the absolute-maximum ratings.

10.9.23. Thermal or Power Ratings. The thermal or power ratings of a semiconductor device may be stated in two different ways: The absolute-maximum internal temperature (usually at the junction) of the device may be quoted along with information concerning the internal temperature rise of the device as a function of power dissipation at particular positions in the device under certain environmental conditions (ambient temperature, coolant velocity, heat-sink attachment, etc.). Alternatively, the absolute-maximum continuous power dissipation at a particular position in the device under certain environmental conditions may be quoted.

The former method is more definitive and thereby more generally useful, while the second method is more explicit and therefore more directly useful if the expected environmental conditions match those for which the maximum power dissipation is quoted.

When power dissipation is not continuous (pulse applications), the maximum power dissipation as determined for continuous operation may normally be safely exceeded. The continuous rating may be simply divided by the duty cycle to obtain the pulse rating, provided the period of operation is much shorter than the internal thermal time constant of the semiconductor device and provided other absolute-maximum ratings [e.g., $I_{C(\max)}$ or $V_{CE(\max)}$] are not exceeded.

If the period of operation is *not* much shorter than the internal thermal time constant but *is* much shorter than the external (case and heat sink) thermal time constant, the following formula can be used to obtain the corrected pulse power rating:

$$P_p = \frac{P_c(\Delta T)_{J \text{ to } A}}{(\Delta T)_{J \text{ to } E} + \xi(\Delta T)_{E \text{ to } A}} \tag{10.9.34}*$$

where P_p = pulse power rating
P_c = continuous power rating
ξ = duty cycle
$(\Delta T)_{J \text{ to } A}$ = temperature difference between the internal point and the ambient for a constant power
$(\Delta T)_{J \text{ to } E}$ = temperature difference between the internal point and an external point for a constant power
$(\Delta T)_{E \text{ to } A}$ = temperature difference between the external point and the ambient for a constant power

Caution must be exercised when designing pulse circuits to assure operation at all times within appropriate maximum ratings. In many cases, electrical ratings rather than thermal ratings will be the determining factors.

10.9.24. Electrical Ratings. Electrical ratings of semiconductor devices include maximum terminal-to-terminal voltages and terminal currents.

Junction breakdown voltages in semiconductor devices are commonly very sharp and the device often exhibits negative-resistance effects in the breakdown region. As a result, if the breakdown voltage is exceeded without a series current-limiting resistance, permanent damage to the device usually results. Generally, absolute-maximum collector and emitter reverse voltages are based on the minimum expected collector and emitter junction breakdown voltages, respectively.

The absolute-maximum collector-to-emitter voltage is determined by the minimum expected reach-through (punch-through) voltage, or the minimum expected collector breakdown voltage, whichever is lower, for the type of device being considered.

The absolute-maximum terminal currents are generally determined by the fuse characteristics of the internal lead wire and should be supplemented by recommended-maximum ratings which indicate the limiting terminal currents at which some other maximum rating, such as power dissipation or junction temperature, may be exceeded.

* It should be noted that this formula is useful only for the case when the period of pulse operation is between the internal and external device thermal time constants.

10.9.25. Mechanical Ratings. The mechanical ratings of semiconductor devices include constant acceleration (centrifuge), vibration, vibration fatigue, shock, and other environmental conditions such as salt atmosphere and high humidity.

The absolute-maximum constant acceleration for a semiconductor device is normally determined by the use of a centrifuge and is rated in units of acceleration of gravity g.

Vibration ratings normally include a frequency range with a constant acceleration at high frequencies and a constant displacement at low frequencies.

Vibration-fatigue ratings indicate the ability of the device to withstand long periods of constant-frequency vibration.

Shock ratings normally include shock duration and shape as well as amplitude and may include information concerning the maximum microphonism of the device for a certain mechanical shock.

10.9.26. Discussion and Comparison of Different Types of Transistors. Many different types of transistors are commercially available or are finding wide application. For purposes of discussion they are conveniently divided into two classes: low-alpha (alpha less than unity) and high-alpha ($\alpha > 1$). Grown-junction, alloy-junction, surface-barrier, intrinsic-barrier, diffused-base, and field-effect transistors are in the $\alpha < 1$; while point-contact, hook-collector, p-n-p-n, double-base-diode, and avalanche transistors exhibit $\alpha > 1$. In the discussion that follows an attempt will be made to point out the distinguishing features of each type of transistor. Article 10.9.27 considers transistors with alpha less than unity and Art. 10.9.28 considers transistors with alpha greater than unity.

10.9.27. "Alpha Less Than Unity" Transistors. (*a*) *Grown Junction.*[65,76] Grown-junction transistors are made from single-crystal germanium which has been grown in such a way that the junctions are "built-in." Small bars are cut from the parent crystal and leads are attached to the three regions. The conductivities of the three regions and the thickness of the center are desired characteristics. The principal advantage of grown-junction transistors lies in the low collector capacitance, which is attributable to the grading of the collector junction. This also results in high collector breakdown voltage and low avalanche multiplication. The alpha may exceed unity by a slight amount at high temperature and high current because of drift fields set up in the collector region which allow additional minority carriers in the collector body to diffuse to the collector junction. Grown-junction transistors have two principal disadvantages in switching circuits: first, the emitter and collector regions may have appreciable resistance which is essentially in series with the closed switch; second, in region III the collector junction injects minority carriers into the collector region which can result in extremely long storage times. The only grown-junction transistors made in significant numbers have been n-p-n transistors. The geometry of a grown-junction transistor is shown in Fig. 10.46*a*.

(*b*) *Tetrode.*[77] To reduce the extrinsic base resistance of grown-junction transistors a second base lead may be added as shown in Fig. 10.46*b*. This results in a device which has been called the junction tetrode. A bias current between the two base leads effectively biases off a large part of the emitter junction, and hence the active part of the transistor is restricted to a small area in the vicinity of one of the base leads. These devices are principally used in common-base circuits and have not found wide application in switching.

(*c*) *Alloy Junction.*[27,57,33,29] Alloyed-junction transistors are fabricated by alloying two metallic contacts on opposite sides of a thin wafer which comprises the base (see Figs. 10.46*c* and 10.36). This contact contains the appropriate donor or acceptor impurity to produce either n-p-n or p-n-p transistors. Since the emitter and collector regions are essentially metallic, alloyed-junction transistors can have extremely low voltage drop collector to emitter in region III. Minority carriers can be stored only in the base region and as a consequence the storage time is reasonably short. The junctions are step junctions in that the resistivity changes rapidly from, say, 0.5 to 5 ohm-cm in the base region to 0.01 to 0.001 ohm-cm in the collector and emitter regions. These step junctions are characterized by a greater capacitance per unit area than graded junctions and by greater avalanche multiplication at the same volt-

age. In practice it is found that alloyed-junction transistors can be made with lower base resistance than grown-junction transistors, which compensates to some extent for the higher collector capacitance in linear applications.

(*d*) *Surface Barrier.*[6] Surface-barrier transistors are similar to alloyed-junction transistors in that the emitter and collector regions are metallic (see Fig. 10.46*d*). These devices are fabricated by etching the base wafer to the thickness desired for alpha and frequency cutoff reasons and then plating on the emitter and collector contacts. Low-resistivity base material is used to prevent punch-through. The contacts

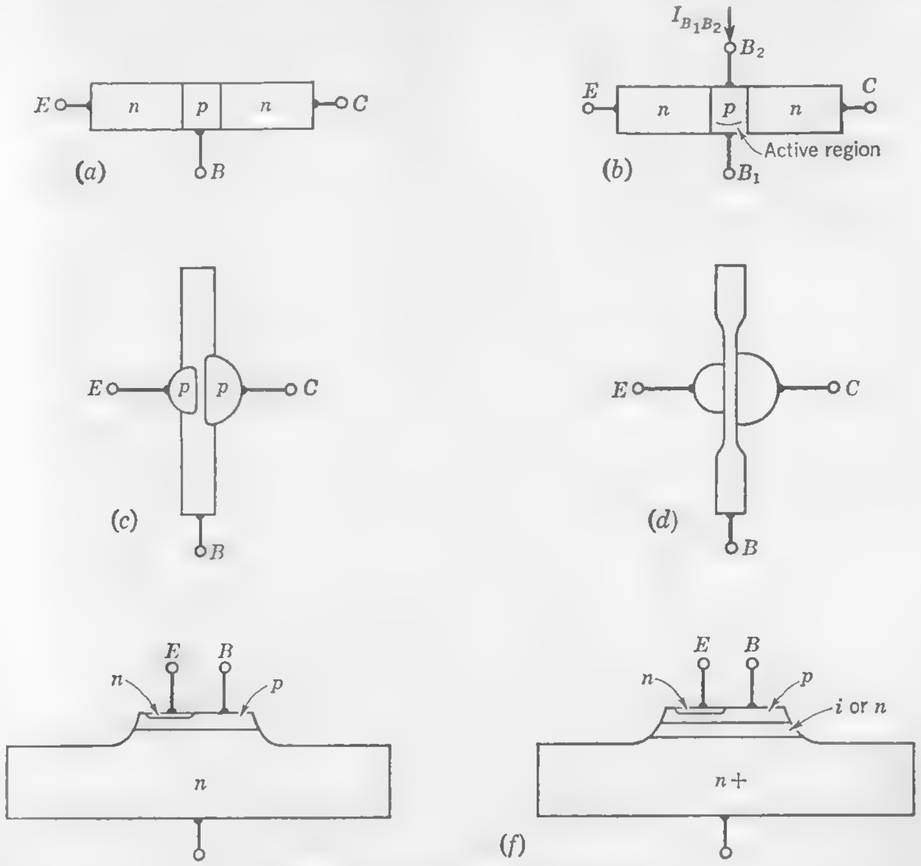


FIG. 10.46*a*. Grown-junction transistor.
FIG. 10.46*b*. Grown-junction tetrode transistor.
FIG. 10.46*c*. Alloy-junction transistor.
FIG. 10.46*d*. Surface-barrier transistor.
FIG. 10.46*e*. Diffused-junction (mesa) transistor.
FIG. 10.46*f*. Intrinsic-barrier diffused-junction transistor.

are small to give low junction reverse current and low capacitance. The junctions result from an inversion layer near the semiconductor surface under the metal contacts. The principal disadvantages are low operating voltage due to the low base-layer resistivity and low power-dissipating ability due to the very small geometry. Circuitwise, surface barrier transistors behave similarly to alloyed-junction transistors but the frequency cutoffs of alpha are possibly a factor of 10 higher because of the thin base region, which results in higher switching speed.

(*e*) *Diffused Junction.*[32,72,34] The diffused-junction (mesa) transistor shown in Fig. 10.46*e* makes use of a thin high-conductivity base layer to obtain high frequency and fast switching response. In this device the thin base layer is obtained by solid diffusion of the appropriate impurity atoms. The use of relatively low resistivity material for the collector region results in a low collector body resistance and hence

makes this transistor appropriate for many switching applications. Furthermore, the frequency cutoff of alpha remains high for low collector voltages and the device may be switched from current cutoff to current saturation at high speeds. Values of alpha cutoff frequency in excess of 1,000 Mc per sec are obtainable.

(*f*) *Intrinsic Barrier.*[15] To reduce the collector body resistance and collector capacitance and to increase the possible collector operating voltage, a transistor, called the intrinsic-barrier transistor, has been developed. The intrinsic or high-resistivity layer is sandwiched between the base layer and the collector body. As conceived, the device also uses a thin high-conductivity base layer which results in high-frequency cutoff and low base resistance (see Fig. 10.46*f*). As a result of the

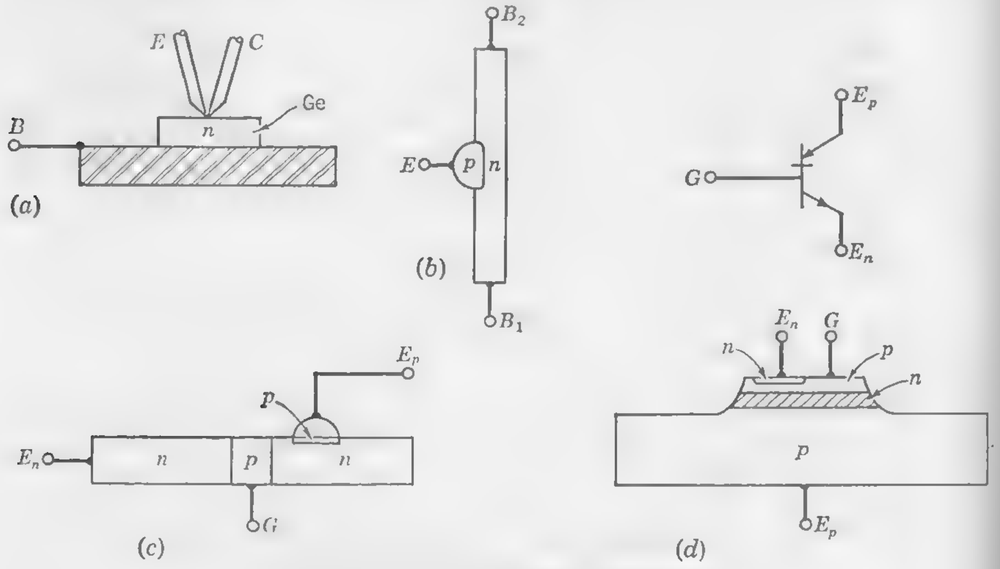


Fig. 10.47*a*. Point-contact transistor.
Fig. 10.47*b*. Unijunction or double-base-diode transistor.
Fig. 10.47*c*. *p-n-p-n* or hook-collector transistor (grown-junction type).
Fig. 10.47*d*. *p-n-p-n* or hook-collector transistor (diffused-junction type).

high operating voltage possible, the power rating of this device may be significantly higher than either a grown- or alloyed-junction transistor.

(*g*) *Field Effect.*[60,12] A field-effect transistor operates in a manner somewhat similar to that of a vacuum tube. The input voltage provides an internal electric field which controls the output current by pinching off the majority-carrier current path through the high-resistivity semiconductor material. The switching speed of the field-effect transistor is limited by transit time and capacitance effects.

10.9.28. "Alpha Greater Than Unity" Transistors. Alpha greater than unity transistors are of particular interest because, with a single transistor, it is possible to design monostable, bistable, and astable pulse circuits with a minimum of additional components.

(*a*) *Point Contact.*[4,3] Point-contact transistors (see Fig. 10.47*a*) are characterized by an alpha or current gain, emitter to collector, greater than unity. This results from a current-multiplying mechanism at the collector contact. Several multiplying mechanisms have been proposed, one of the most likely being that the collector multiplication is a result of conductivity modulation (see Art. 10.8.3) in the collector region. It has also been proposed that the collector region (including the collector-point semiconductor contact) is in effect a double junction or "hook." Possibly, the true mechanism is a combination of these two. In order to obtain multiplication of emitted current it is necessary to "form" the collector point, which means to pulse the collector with a high current. Even though the exact mechanism of operation of the point-contact transistor is not completely understood, it has been possible to design them to

meet the requirements of several applications and to fabricate them with reproducible characteristics and with good life and reliability. By fabricating units with close point spacing, point-contact transistors have been produced in quantity to operate in digital-computer circuits at as high as 3 Mc pulse rates.[68,66]

(*b*) *Unijunction or Double-base Diode.*[35,64] An additional type of semiconductor device which is characterized by an alpha greater than unity is the unijunction or double-base diode or filamentary transistor. This device is shown in Fig. 10.47*b*. It is composed of two ohmic contacts on the ends of a semiconductor filament and an emitter of minority carriers along the bar. Such a device may be used as an amplifier or as the active element in a bistable circuit. The speed is limited by transit-time effects.

(*c*) *p-n-p-n or Hook Collector.*[65,16,46,41] The *p-n-p-n* or hook-collector transistor structures shown in Figs. 10.47*c* and *d* represent grown-junction and diffused-junction

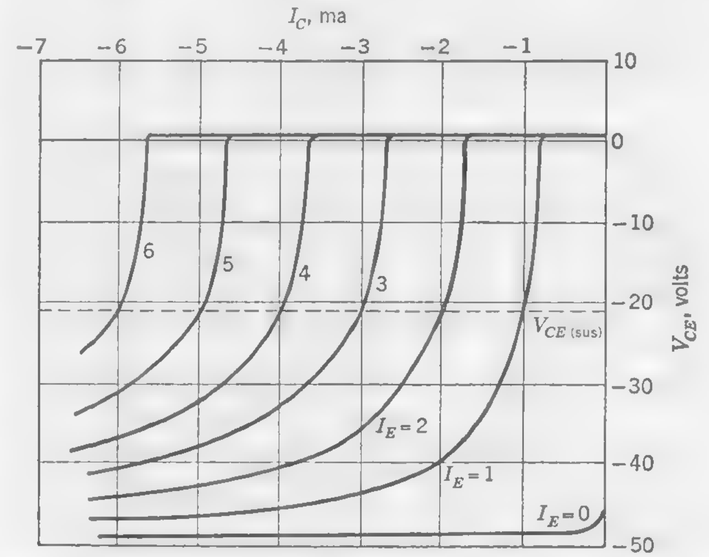


Fig. 10.48. Common-base collector characteristic of an avalanche transistor.

technologies, respectively. The electron current emitted by the *n*-emitter (E_n) diffuses across the adjacent *p* region and is collected at the central *p-n* junction. This current forward-biases the *p*-emitter (E_p) where holes are injected into the middle *n* region; they diffuse across this region and are collected at the central *p-n* junction. This collected current causes additional electron emission from the *n*-emitter which is collected, and results in increased hole emission from the *p*-emitter, etc. The process is regenerative and represents a transistor current gain which is greater than unity. The transistor is triggered from OFF to ON state by applying a gate current to forward-bias the *n*-emitter. It is turned OFF by reducing the emitter-to-emitter ($E_n - E_p$) current below the holding current where the effective current gain is less than unity. *p-n-p-n* or hook-collector transistors are available in a wide variety of types operable at currents ranging from less than 1 ma to more than 50 amp with switching times less than 100 *n*sec.

(*d*) *Avalanche.*[43,30,42] Transistors have been made which emphasize avalanche multiplication to give an alpha greater than unity. These avalanche transistors are of the alloy type and have static collector characteristics as shown in Fig. 10.48. At a voltage

$$V_{CE(\text{sus})} = BV^n \sqrt{1 - \bar{\alpha}_0} \tag{10.9.35}$$

called the sustaining voltage, the alpha of such a device becomes equal to unity, $\bar{\alpha}_0$ being the alpha at low voltage and BV' being the junction breakdown voltage. The

variation of alpha with voltage is given by the relation

$$\bar{\alpha} \approx \frac{\bar{\alpha}_0}{1 - (V_C/BV)^n} \tag{10.9.36}$$

$\bar{\alpha}$ approaches infinity as the collector voltage approaches BV. For alloyed-junction p-n-p germanium transistors the parameter n is approximately 3. Hence for $\bar{\alpha}_0 = 0.96$, the alpha is greater than unity for voltages greater than about one-third the breakdown voltage. Whereas these devices have not been exploited to any great extent, they are capable of switching at megacycle rates. They may be used in astable, bistable, or monostable switching or pulse circuits. Their main disadvantage is a relatively high ON state voltage drop.

10.9.29. Typical Transistor Characteristics. More than one thousand transistor codes are commercially available. These include germanium point-contact and both germanium and silicon grown-junction, alloy-junction, and diffused-junction types. The electrical characteristics of typical switching transistors from each of these classes of devices are given in Table 10.9. The parameter values given in the table are representative of the indicated transistor class and are considered to be conservative. By designing devices specifically for a given application, improvements may be possible in certain critical parameters.

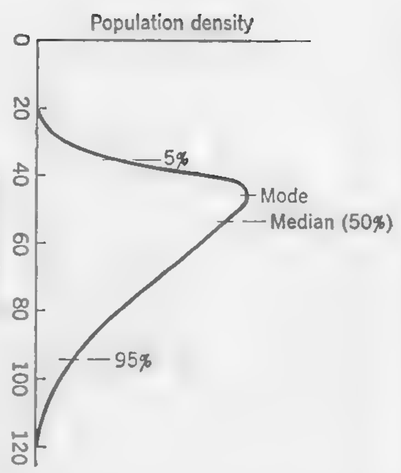


FIG. 10.49. Distribution of current gain (h_{FE}) for a typical universe of diffused-silicon n-p-n transistors.

10.9.30. Variation of Parameters. (*a*) *Parameter Distributions.* Electronic devices manufactured by a given process can be expected to have different electrical characteristics. The width of the range for a given parameter depends upon the accuracy with which the various process variables are controlled. If the density of units is plotted against a given parameter for a large number of units, it would be expected that this parameter distribution would resemble the normal probability curve. In actual practice the curve may be skewed (see Fig. 10.49) or even exhibit two peaks (bimodal). In order to indicate the typical distribution the device manufacturer may give the median value and two other values to indicate the range; for example, the 5 and 95 percentiles.

(*b*) *Variation with Operating Point and Temperature.* Transistor parameters vary with operating point and temperature. Junction reverse currents usually double every 8 to 10°C; however, if a considerable portion of the reverse current represents leakage current, it may double only every 15°C. Because of space-charge-layer widening, alpha and alpha cutoff frequency tend to increase with collector voltage in junction transistors. These parameters also vary with emitter current because of drift fields in the base region, variation of emitter efficiency, and difference in flow paths. Typical variation of current gain with emitter current[79] and temperature is shown in Fig. 10.50 for a diffused silicon n-p-n transistor. The extent to which parameter variation with operating conditions must be studied and specified depends, of course, on the type of application.

(*c*) *Manufacturer, Data Sheet, and Circuit Limits.* Frequently, not all the devices are marketable and it is necessary to establish test limits to limit the distribution. For some parameters, such as I_{CO}, only an upper limit is necessary (see Fig. 10.51). In order to assure that units do not drift out of manufacturer's limits, which appear on the data sheets, manufacturer's limits may be established inside the data-sheet requirements. To assure that devices with acceptable characteristics are used in a given application and to allow for parameter aging, circuit-requirement limits should be outside the data-sheet limits. Finally, end-of-life limits are determined to specify

Table 10.9. Typical Limiting Characteristics of Various Transistor Types

Transistor type	Absolute max power at 25°C, mw	Absolute max V_{OFF}, volts	Recommended max I_{ON}, ma	Max V_{ON} at I_{ON}		Max I_{OFF}, μa	Min $\bar{\alpha}_f$, —	Min $f_{\alpha f}$, Mc	Max C_C at $V_R = 5V$, pf	Max $C_C + C_E$ at OV, pf	Max τ_r for $\omega_{\alpha f} R_C C_C \ll 1$, μsec	Max τ_s, μsec
				Volts	Ma							
Grown junction:												
Ge n-p-n	120	35	30	2.0	10	15	0.96	1	15	300	10.0	20
Si n-p-n	125	45	20	1.5	10	1	0.95	4		80	3.0	4
Alloy junction:												
Ge n-p-n	250	35	200	0.075	10	10	0.98	4	30	250	3.0	1.0
Ge p-n-p	250	50	200	0.075	10	10	0.98	4	25	250	3.0	0.7
Ge p-n-p (high I)	2,500	40	1,000	0.250	500	15	0.95	4	45	550	2.5	1.0
Si p-n-p	150	10	50	0.150	5	0.1	0.94	5	15	40	2.0	0.8
Diffused junction:												
Ge p-n-p (mesa)	150	20	50	0.3	10	1.0	0.96	600	3	18	0.03	0.03
Ge p-n-p (mico alloy)	60	15	50	0.2	10	5	0.95	200	5	30	0.08	0.04
Ge p-n-p (high I mesa)	6,000	35	300	3.0	200	15	0.95	400	15	180	0.05	0.80‡
Si n-p-n (mesa)	250	25	100	0.2	10	0.1	0.97	300	5	30	0.06	0.02
Si n-p-n (high I mesa)	12,500	100	2,000	1.0	1,000	0.1	0.96	60	50	300	0.25	0.10
$\alpha > 1$:												
Ge point contact	200	50	40	2.5	18	2,500	1.8	2.0	2	20	†	†
Si p-n-p-n	600	400	5,000*	1.0	100	1	†	†	5	†	†	†
Si unijunction	450	60	2,000*	4.3	50	2	†	†	†	†	†	†

* Pulse.
† Not applicable.
‡ Intrinsic barrier (epitaxial) design should reduce this τ_s by about a factor of ten.

the end of the useful life in the given application since it is recognized that most all electronic devices may deteriorate because of chemical and physical changes. A device is listed as a failure if it is no longer operable in the circuit under all reasonable, and to be expected, environmental conditions.

10.9.31. Reliability. A device is considered to be reliable if it performs its intended function for a satisfactory length of time. Thus, a given type of electronic device

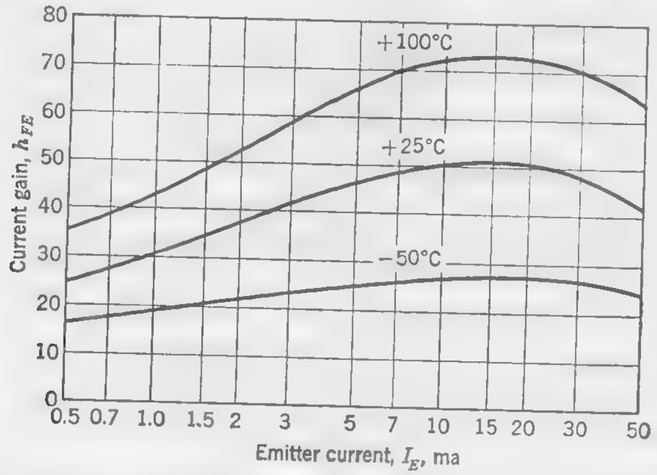


FIG. 10.50. Variation of current gain (h_{FE}) with emitter current and temperature for a typical low power diffused-silicon n-p-n transistor.

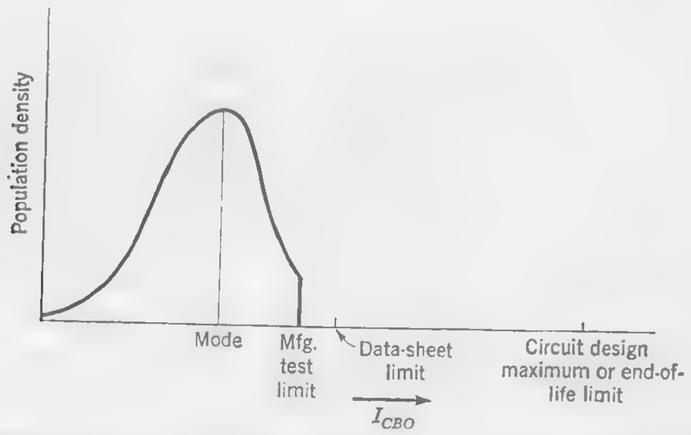


FIG. 10.51. Distribution of cutoff current (I_{CBO}) for a universe of transistors, showing manufacturing-test limit, data-sheet or specification limit, and the design maximum or end-of-life limit.

may prove completely reliable in one application and at the same time exhibit totally inadequate life performance in another application.

Much can be accomplished by the circuit and system designer to improve tolerance to existing device failure rates. Circuits can be designed with large safety margins to permit major device degradation before circuit failure occurs, and periodic marginal checking can be performed to catch potential degradation-type device failures and thereby prevent possible system failures. In addition, circuit redundancy can be used to prevent system failure when catastrophic-type device failures occur. It must be recognized that these circuit and system design procedures normally degrade circuit and system performance and increase system cost. As a result, when evolving a system design having satisfactory life performance, the circuit and system engineer

must balance the cost of these procedures against the cost of devices having the required reliability assurance.

(a) *Types of Failure, Catastrophic, Degradation—Failure Rate.* Device reliability is generally referred to in terms of device failure rate under certain specified electrical and environmental conditions. Failures of electronic devices can be broadly classified as being due to either catastrophe or degradation. Examples of catastrophic failures are interelectrode shorts, broken leads, or any such occurrence which results in sudden and complete failure, previous to which time the device gave no indication of impending breakdown. Certain types of transistors have shown a sudden decrease in breakdown voltage. Gradual deterioration and subsequent failure, on the other hand, are a slow degradation of a parameter or parameters with time. The mechanisms which result in gradual deterioration are physical or chemical in nature, and the deterioration usually follows an exponential law. Such parameter changes may be temperature-dependent. Both types of aging are shown in Fig. 10.52. The curves

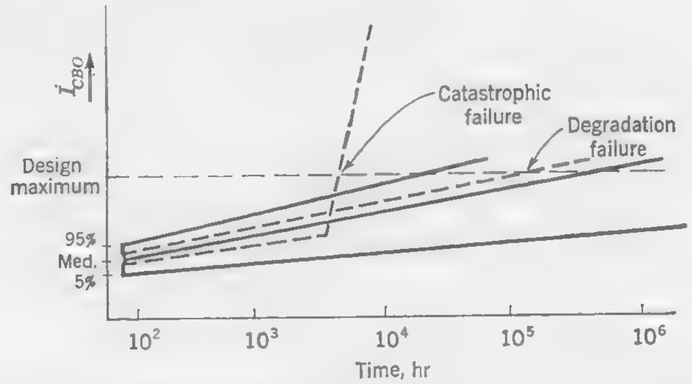


FIG. 10.52. Exemplary aging behavior of a group of electronic devices.

in this figure indicate the aging of the median of the group as well as the variation of the range (note spreading of the 5 and 95 percentiles.) A device which exhibits sudden or catastrophic failure is indicated by the dotted line. An end-of-life limit is also shown which indicates that the average life is about 80,000 hr under the specified operating conditions. A preferable way of reckoning device life is to determine the failure rate. The failure rate in per cent per 1,000 hr is given by

$$\text{F.R.} = \frac{\text{no. of failures} \times 10^5}{\text{total transistor socket hr}} \tag{10.9.37}$$

Thus, if in a system containing 200 transistors there are two failures in 10,000 hr, the failure rate is 0.1 per cent per 1,000 hr.

(b) *Device Reliability (Failure-rate) Requirements.* The allowable device failure rate for a given circuit and system application can only be determined by performing a system-reliability study. This is commonly achieved by combining the expected failure rates for each device type, in proportion to the number of devices of each type, with the expected failure rates of other components, connections, etc., in order to determine the expected over-all system failure rate. This determination is then compared with the required system failure rate to determine adequacy. A reiterative procedure can be used to determine reasonable individual device failure-rate requirements to optimize system cost and performance factors.

(c) *Device Reliability Measurement and Specification.* Device reliability is commonly measured by conducting life tests under accelerating conditions, such as high temperature, current, voltage, or power, which are relatable to those actually to be encountered in expected applications. The relationship between condition stress level and device failure rate is normally very close to log linear. If the dominant failure mechanism is chemical in nature, a better log-linear fit will usually be obtained

to the reciprocal of absolute temperature rather than directly to temperature. Figure 10.53 shows a typical transistor failure-rate characteristic as a function of temperature. The required device reliability may be assured by specifying suitable maximum failure rates for device life tests which are performed on adequate samples which represent the devices to be used in the system being considered. These tests are normally performed by the device manufacturer since large numbers of devices are usually required and he can use the results of each life test to certify the reliability of devices for many system applications.

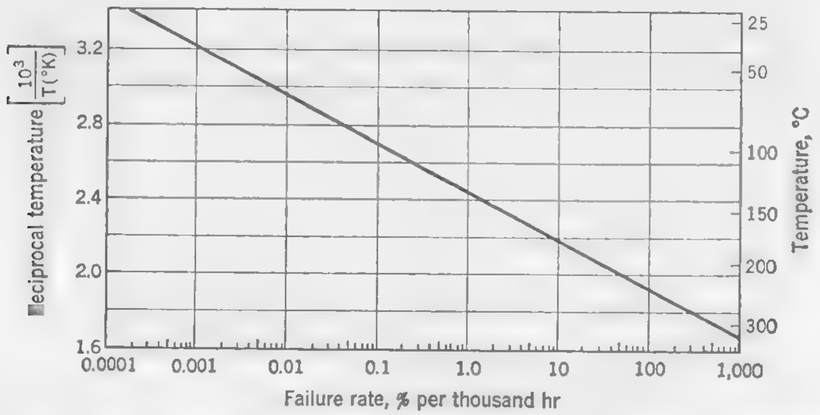


FIG. 10.53. Typical transistor failure-rate characteristic as a function of temperature.

References

1. Anderson, A. E.: Transistors in Switching Circuits, *Proc. IRE*, **40**:1541–1558 (November, 1952).
2. Anderson, J. R.: Electrical Delay Lines for Digital Computer Applications, *IRE Trans. on Electronic Computers*, **EC-2** (2): 5–13 (June, 1953).
3. Bardeen, J., and W. H. Brattain: Physical Principles Involved in Transistor Action, *Phys. Rev.*, **75**:1208–1225 (Apr. 15, 1949).
4. Bardeen, J., and W. H. Brattain: The Transistor, a Semiconductor Triode, *Phys. Rev.*, **74**:230–231 (July, 1948).
5. Bemski, G.: Recombination in Semiconductors, *Proc. IRE*, **46**:99–1004 (June, 1958).
6. Bradley, W. E., J. W. Tiley, R. A. Williams, J. B. Angell, F. P. Keiper, Jr., R. Kansas, R. F. Schwarz, and J. F. Walsh: The Surface Barrier Transistor, *Proc. IRE*, **41**:1702–1720 (December, 1953).
7. Bray, R., and B. R. Gossick: Effect of Transit Time on Germanium Rectifier Behavior, *Phys. Rev.*, **91**:1011–1012 (Aug. 15, 1953).
8. Bright, R. L.: Junction Transistors Used as Switches, *Trans. AIEE*, **74** (part I): 111–121 (March, 1955).
9. Brown, W. L.: *N*-type Surface Conductivity on *P*-type Germanium, *Phys. Rev.*, **91**: 518–527 (Aug. 1, 1953).
10. Chance, B., V. Hughes, E. F. MacNichol, D. Sayre, and F. C. Williams: *Waveforms* McGraw-Hill Book Company, Inc., New York, 1949.
11. Credle, A. B.: Effects of Low Temperatures on Transistor Characteristics, *IBM J. Research Development*, **2** (1): 54–71 (January, 1958).
12. Dacey, G. C., and I. M. Ross: Unipolar Field-effect Transistor, *Proc. IRE*, **41**:970–979 (August, 1953).
13. Early, J. M.: Design Theory of Junction Transistors, *Bell System Tech. J.*, **32**:1271–1312 (November, 1953).
14. Early, J. M.: Effects of Space-charge Layer Widening in Junction Transistors, *Proc. IRE*, **40**:1401–1406 (November, 1952).
15. Early, J. M.: *p-n-i-p* and *n-p-i-n* Junction Transistor Triodes, *Bell System Tech. J.*, **33**:517–533 (May, 1954).
16. Ebers, J. J.: Four-terminal, *p-n-p-n* Transistors, *Proc. IRE*, **40**:1361–1364 (November, 1952).

17. Ebers, J. J., and S. L. Miller: Design of Alloyed Junction Germanium Transistors for High-speed Switching, *Bell System Tech. J.*, **34**:761–781 (July, 1955).
18. Ebers, J. J., and J. L. Moll: Large-signal Behavior of Junction Transistors, *Proc. IRE*, **42**:1761–1772 (December, 1954).
19. Farley, B. G.: Dynamics of Transistor Negative-resistance Circuits, *Proc. IRE*, **40**: 1497–1508 (November, 1952).
20. Firle, T. E., M. E. McMahon, and J. F. Roach: Recovery Time Measurements on Point-contact Germanium Diodes, *Proc. IRE*, **43**:603–607 (May, 1955).
21. Gano, J. J., and G. F. Sandy: Thermistors for the Gradual Application of Heater Voltage to Thermionic Tubes, *IRE Trans. on Electronic Computers*, **EC-7** (1): 61–64 (March, 1958).
22. Giacoletto, L. J.: Junction Transistor Equivalent Circuits and Vacuum Tube Analogy, *Proc. IRE*, **40**:1490–1493 (November, 1952).
23. Giacoletto, L. J.: Study of *p-n-p* Alloy Junction Transistor from dc through Medium Frequencies, *RCA Rev.*, **15**:506–562 (December, 1954).
24. Glasoe, G. N., and J. V. Lebacqz: *Pulse Generators*, part III, McGraw-Hill Book Company, Inc., New York, 1948.
25. Goucher, F. S., *et al.*: Theory and Experiment for a Germanium *p-n* Junction, *Phys. Rev.*, **81**:637–638 (Feb. 15, 1951).
26. Hall, R. N.: Germanium Rectifier Characteristics, *Phys. Rev.*, **83**:228 (July 1, 1951).
27. Hall, R. N., and W. C. Dunlap: *P-N* Junctions Prepared by Impurity Diffusion, *Phys. Rev.*, **80**:467–468 (Nov. 1, 1950).
28. I.R.E. Standard on Solid-state Devices: Methods of Testing Transistors, *Proc. IRE*, **44**:1542–1561 (November, 1956).
29. Jenny, D. A.: A Germanium *n-p-n* Alloy Junction Transistor, *Proc. IRE*, **41**:1728–1734 (December, 1953).
30. Kidd, M. C., W. Hasenberg, and W. M. Webster: Delayed Collector Conduction, a New Effect in Junction Transistors, *RCA Rev.*, **16**:16–33 (March, 1955).
31. Kingston, R. H.: Switching Time in Junction Diodes and Junction Transistors, *Proc. IRE*, **42**:829–834 (May, 1954).
32. Krömer, H.: Der Drifttransistor, *Naturwiss.*, **40**:578–579 (November, 1953).
33. Law, R. R., C. W. Mueller, J. I. Pankove, and L. D. Armstrong: A Developmental Germanium *p-n-p* Junction Transistor, *Proc. IRE*, **40**:1352–1357 (November, 1952).
34. Lee, C. A.: A High-frequency Diffused Base Germanium Transistor, *Bell System Tech. J.*, **35**:23–34 (January, 1956).
35. Lesk, I. A., and V. P. Mathis: The Double-base Diode: A New Semiconducting Device, *IRE Convention Record*, part 6, New York, pp. 2–8, 1953.
36. Lesk, I. A., N. Holonyak, Jr., and U. S. Davidsohn: The Tunnel Diode—Circuits and Applications, *Electronics:* 60–64 (Nov. 27, 1959).
37. Lo, A. W.: Transistor Trigger Circuits, *Proc. IRE*, **40**:1531–1541 (November, 1952).
38. MacFadyen, K. A.: *Small Transformers and Inductors*, Chapman & Hall, Ltd., London, 1953.
39. McAfee, K. B., E. J. Ryder, W. Shockley, and M. Sparks: Observations of Zener Current in Germanium *p-n* Junctions, *Phys. Rev.*, **83**:650–651 (Aug. 1, 1951).
40. McKay, K. G.: Avalanche Breakdown in Silicon, *Phys. Rev.*, **94**:877–884 (May, 1954).
41. Mackintosh, I. M.: The Electrical Characteristics of Silicon *p-n-p-n* Triodes, *Proc. IRE*, **46**:1229–1235 (June, 1958).
42. Miller, S. L.: Avalanche Breakdown in Germanium, *Phys. Rev.*, **99**:1234–1241 (August, 1955).
43. Miller, S. L., and J. J. Ebers: Alloyed Junction Avalanche Transistors, *Bell System Tech. J.*, **34**:883–902 (September, 1955).
44. Millman, J., and H. Taub: *Pulse and Digital Circuits*, McGraw-Hill Book Company, Inc., New York, 1956.
45. Moll, J. L.: Large Signal Transient Response of Junction Transistors, *Proc. IRE*, **42**:1773–1784 (December, 1954).
46. Moll, J. L., M. Tanenbaum, J. M. Goldey, and N. Holonyak: *p-n-p-n* Transistor Switches, *Proc. IRE*, **44**:1174–1182 (September, 1956).
47. Nanavati, R. P.: Prediction of Storage Time in Junction Transistors, *IRE Trans. on Electron Devices*, **ED-7** (1): 9–15 (January, 1960).
48. Pearson, G. L., and W. H. Brattain: History of Semiconductor Research, *Proc. IRE*, **43**:1794–1806 (December, 1955).
49. Pearson, G. L., and C. S. Fuller: Silicon *p-n* Junction Power Rectifiers and Lightning Protectors, *Proc. IRE*, **42**:760 (April, 1954).
50. Pearson, G. L., and B. Sawyer: Silicon *p-n* Junction Alloy Diodes, *Proc. IRE*, **40**: 1348–1351 (November, 1952).

51. Peterson, L. C.: Equivalent Circuits of Linear Active Four Terminal Networks, *Bell System Tech. J.*, **27**:593–622 (October, 1948).
52. Pietenpol, W. J.: *P-N* Junction Rectifier and Photocell, *Phys. Rev.*, **82**:120–121 (Apr. 1, 1951).
53. Pritchard, R. L.: Frequency Variations of Current Amplification Factor for Junction Transistors, *Proc. IRE*, **40**:1476–1481 (November, 1952).
54. Reintjes, J. F., and G. F. Coate: *Principles of Radar*, McGraw-Hill Book Company, Inc., New York, 1952.
55. Rouault, C. L., and G. N. Hall: A High-voltage Medium Power Rectifier, *Proc. IRE*, **40**:1519–1521 (November, 1952).
56. Ryder, R. M., and R. J. Kircher: Some Circuit Aspects of the Transistor, *Bell System Tech. J.*, **28**:367–401 (July, 1949).
57. Saby, J. S.: Fused Impurity *p-n-p* Junction Transistors, *Proc. IRE*, **40**:1358–1360 (November, 1952).
58. Schenkel, H., and H. Statz: Junction Transistors with Alpha Greater Than Unity, *Proc. IRE*, **44**:360–371 (March, 1956).
59. Schulman, R. G., and M. E. McMahon: Recovery Currents in Germanium *p-n* Junction Diodes, *J. Appl. Phys.*, **24**:1267–1272 (October, 1953).
60. Shockley, W.: A Unipolar "Field-effect" Transistor, *Proc. IRE*, **40**:1365–1376 (November, 1952).
61. Shockley, W.: *Electrons and Holes in Semiconductors*, D. Van Nostrand Company, Inc., Princeton, N.J., 1950.
62. Shockley, W.: The Theory of *p-n* Junctions in Semiconductors and *p-n* Junction Transistors, *Bell System Tech. J.*, **28**:435–489 (July, 1949).
63. Shockley, W.: Transistor Electronics—Imperfections, Unipolar and Analog Transistors, *Proc. IRE*, **40**:1289–1313 (November, 1952).
64. Shockley, W., G. L. Pearson, and J. R. Haynes: Hole Injection in Germanium; Quantitative Studies and Filamentary Transistors, *Bell System Tech. J.*, **28**:344–366 (July, 1949).
65. Shockley, W., M. Sparks, and G. K. Teal: *P-N* Junction Transistors, *Phys. Rev.*, **83**:151–162 (July 1, 1951).
66. Simkins, Q. W., and J. H. Vogelsong: Transistor Amplifiers for Use in a Digital Computer, *Proc. IRE*, **44**:43–55 (January, 1956).
67. Simmons, C. D.: A Practical Method for the Specification of the Transient Response of a Junction Transistor, thesis, University of Pennsylvania, Moore School of Electrical Engineering, June, 1960.
68. Slade, B. N.: The Control of Frequency Response and Stability of Point-contact Transistors, *Proc. IRE*, **40**:1382–1384 (November, 1952).
69. Sparkes, J. J.: A Study of the Charge Control Parameters of Transistors, *Proc. IRE*, **48** (10): 1696–1705 (October, 1960).
70. Steele, E. L.: Theory of Alpha for *p-n-p* Diffused Junction Transistors, *Proc. IRE*, **40**: 1424–1428 (November, 1952).
71. Strickland, P. R.: The Thermal Equivalent Circuit of a Transistor, *IBM J. Research Development*, **3** (1): 35-45 (January, 1959).
72. Tanenbaum, M., and D. E. Thomas: Diffused Emitter and Base Silicon Transistors, *Bell System Tech. J.*, **35**:1–22 (January, 1956).
73. Thomas, D. E.: Transistor Amplifier Cutoff Frequency, *Proc. IRE*, **40**:1481–1483 (November, 1952).
74. Torrey, H. C., and C. A. Whitmer: *Crystal Rectifiers*, McGraw-Hill Book Company, Inc., New York, 1948.
75. Von Tersch, L. W., and A. W. Swago: *Recurrent Electrical Transient*, sec. 2–3, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1953.
76. Wallace, R. L., Jr., and W. J. Pietenpol: Some Circuit Properties and Applications of *n-p-n* Transistors, *Proc. IRE*, **39**:753–767 (July, 1951). Also *Bell System Tech. J.*, **30**:530–563 (July, 1951).
77. Wallace, R. L., Jr., L. G. Schimpf, and E. Dickten: A Junction Transistor Tetrode for High Frequency Use, *Proc. IRE*, **40**:1395–1400 (November, 1952).
78. Waltz, M. C.: On Some Transients in the Pulse Response of Point-contact Germanium Diodes, *Proc. IRE*, **40**:1483–1487 (November, 1952).
79. Webster, W. M.: On the Variation of Junction-transistor Current-amplification Factor with Emitter Current, *Proc. IRE*, **42**:914–920 (June, 1954).

Section 11

SINGLE-INPUT COMPONENT CIRCUITS

By

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

BRAM J. LOOPSTRA, *N. V. Electrologica, Amsterdam, The Netherlands.*

CONTENTS

11.1. Amplifier Response in Computer Applications 11-1
11.2. Cathode Followers and Emitter Followers 11-2
11.3. D-C Level Restoration 11-4
11.3.1. An Example of D-C Level Arrangement in a Computer 11-4
11.3.2. The Requirements of a D-C Level Restorer 11-4
11.4. Buffer Inverters 11-5
11.4.1. A Transistor Buffer Inverter 11-6
11.5. Relay Drivers 11-7
11.6. Signal-modifying Circuits 11-7
11.6.1. Clipping Circuits 11-8
11.6.2. Attenuation 11-8
11.6.3. Ringing Circuits 11-8
11.6.4. Pulse-train Generator 11-9
11.6.5. Delay Circuits 11-9
11.6.6. Schmidt Triggers 11-11
11.6.7. Pulse-broadening Circuits 11-11
11.6.8. Pulse-generating Circuits 11-13
11.7. Amplifiers 11-13
11.7.1. Introduction 11-13
11.7.2. Distortion 11-15
11.7.3. Noise 11-16
11.7.4. Feedback 11-17
11.7.5. Direct-coupled Amplifiers 11-18
11.7.6. Cathode-coupled Amplifiers 11-24
11.7.7. Regulator Amplifiers 11-24
11.7.8. RC-coupled Amplifiers 11-25
11.7.9. Transformer-coupled Amplifiers 11-33
11.7.10. Tuned Amplifiers 11-34
11.7.11. Transformer-coupled Drivers 11-36
11.8. Parametric Oscillators 11-37

During the preparation of this section there has been a complete change from vacuum-tube circuitry to transistor circuitry in digital computers. Magnetic amplifiers have been used to some extent, but now transistors are being used most extensively.

11.1. AMPLIFIER RESPONSE IN COMPUTER APPLICATIONS

Figure 11.1 shows the idealized response of a large-signal-computer amplifier stage. With signals of less amplitude than B (noise), there is no output. With signals greater than C, there is full standard output. Ideally, the difference in level between B and C is as small as possible. Then substantial variations in input signal cause no change in

output. Such idealized requirements can very nearly be met by the use of cutoff characteristics of vacuum-tube circuits and their saturation properties. It may be necessary to use "clamping" circuits such as those of Fig. 11.6 to keep the signal in the range from −20 volts to 0 volt, or in the case of pulse circuitry to limit the maximum signal to, say, 20 volts.

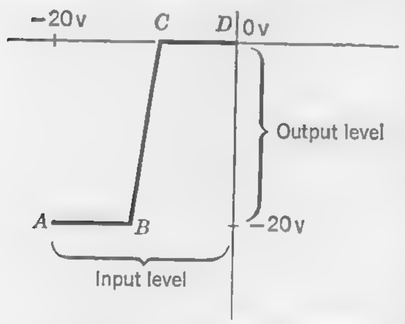


Fig. 11.1. Ideal response.

Some computer designers have discussed the possibility of designing a circuit which detects the presence of any signals falling in the interval between B and C. The more interesting designs would interpret such signals as 1's but would set alarm indicators so that those units could be replaced at the end of the computation. The choice of 1's is due to the fact that signals generally decrease in size with aging of tubes, but noise does not increase. So far, such designs appear to be too complicated and the success of marginal checking techniques has limited further development of these ideas.

Generally, pulse amplifiers must restore the size of the pulse or increase the power. Circuits which do these things will be discussed in the following sections.

11.2. CATHODE FOLLOWERS AND EMITTER FOLLOWERS

The characteristics of the cathode-follower circuit which are of most interest in digital computer applications are the following:

1. High input impedance
2. Low output impedance
3. No inversion of signal

The features present in cathode followers and less attractive to the computer circuit designer are (1) less than unity gain, and (2) the change of d-c level between the input and output. Thus, in computer circuits cathode followers are used for power amplification and in some instances for logical purposes.

In the discussion of cathode followers the following symbols will be used (see also the discussion of cathode followers in Art. 11.7.5 on direct-coupled amplifiers):

C_{gk} = static grid-to-cathode capacitance
C_{gp} = static grid-to-plate capacitance
C_{in} = cathode-follower input capacity
g_m = transconductance of the tube
R_o = output impedance
R_k = cathode load resistance
μ = amplification factor

If R_k is the cathode resistor, then the equations for the gain and the output impedance are respectively

$$\text{Gain} = \frac{g_m R_k}{1 + g_m R_k(\mu + 1/\mu)} \tag{11.1}$$

$$R_o = \frac{R_k}{1 + g_m R_k(\mu + 1/\mu)} \tag{11.2}$$

For the case $\mu \gg 1$, the following equations are used to good approximation:

$$\text{Gain} \cong \frac{g_m R_k}{1 + g_m R_k} \tag{11.3}$$

$$R_o \cong R_k(1 - \text{gain}) \tag{11.4}$$

The input capacitance for a triode cathode follower is

$$C_{in} \cong C_{gp} + (1 - \text{gain})C_{gk} \tag{11.5}$$

and for a pentode with the screen bypassed to cathode is

$$C_{in} \cong (1 - \text{gain})(C_{gs} + C_{gk}) \tag{11.6}$$

where C_{gp} = static grid-to-plate capacitance
C_{gk} = static grid-to-cathode capacitance
C_{gs} = static grid-to-screen capacitance

When a cathode follower is used to drive a capacitance load C, the rise time can be calculated by using Eq. (11.4), namely,

$$\text{Rise time} = 2.2\, R_0 C \tag{11.7}$$

However, for fast input pulse of large amplitude, Eq. (11.7) will not apply for the calculation of fall time of the output pulse. The fast fall time of the input pulse

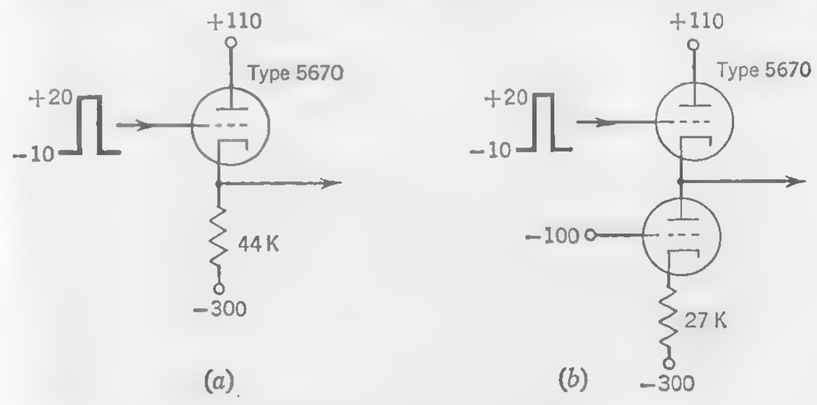


Fig. 11.2. Cathode followers.

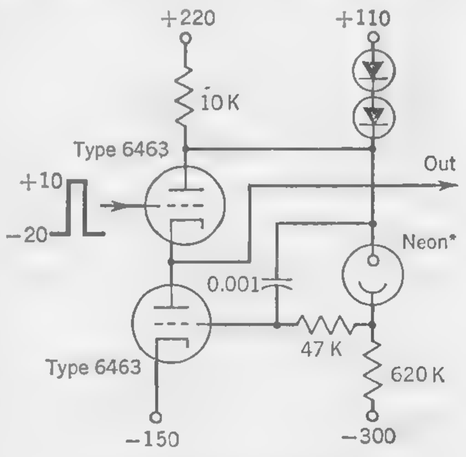
causes the tube to cut off and the fall time must be calculated by the cathode network alone.

In computer applications, frequently many cathode followers are connected in series with logical circuits between stages. If these circuits are direct-coupled, measures must be taken to take care of the cathode-bias rise, which can be considerable if the voltage swing is large. One way to minimize the cathode rise is to replace the cathode resistance by a constant-current source. This may be done by using a large cathode resistance returned to an appropriate voltage source (Fig. 11.2*a*) or by using a constant-current tube in the cathode circuit (Fig. 11.2*b*).

When a d-c-coupled driver is required to drive a large-capacitance load, instead of using cathode followers in parallel, employing feedback in the cathode circuit of a cathode follower sometimes is used to advantage. Figure 11.3 shows such a circuit.

The frequency response of the transistor may cause signal overshoot in circuits such as that shown in Fig. 11.4.

Like the cathode follower, the emitter follower provides power gain, no inversion of signal, and some attenuation and change in d-c level. If the output impedance is made small the gain is nearly unity and the shift in level is small. Figure 11.5 shows an arrangement for the restoration of d-c level.



*Five Ne-2 neons in series.
The neons are aged and selected
at 60 volts drop per neon.

Fig. 11.3. Cathode follower with feedback.

Usually it is desirable to prevent saturation. This can be accomplished by making sure that the supply voltage is more negative than the negative excursions of the signals. Since in these circuits the transistor is in the active region, attention should be given to the power dissipated in the transistor.

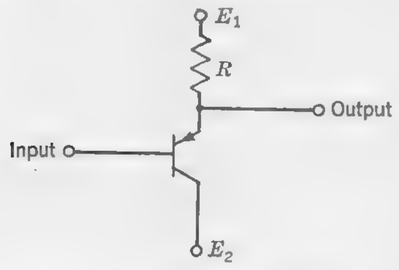


FIG. 11.4. Emitter follower.

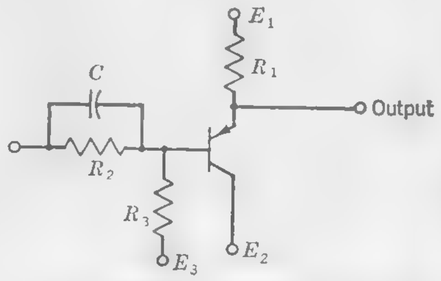


FIG. 11.5. Emitter follower with d-c level change.

11.3. D-C LEVEL RESTORATION

Many signals present in digital computer circuits must exist for long periods of time. For example, they must control the flow of pulses for hundreds of pulse times. This section considers the restoration of signal levels for such signals, whether they are actual d-c signals (maintaining their levels if the computer is halted) or whether they last for only a few pulse times. Loss of level may occur in d-c coupling networks, in logical gating circuitry, or in cathode followers.

11.3.1. An Example of D-C Level Arrangement in a Computer. Suppose that a computer is designed to use binary logic, that is, each circuit is to be either OFF or ON. The ON condition can be represented by a binary digit of 1 and the OFF condition by a binary 0. Suppose further that the OFF condition is represented by a signal level of −20 volts and that the ON condition is represented by a signal level of 0 volt.

After such signals have passed through a chain of logical circuits an ON signal may have deteriorated to, say, −3 volts; or an OFF signal may be only −16 volts. Suppose, ideally, that a restoration circuit is available which with an input greater than −10 volts gives an output of 0, and for an input of less than −10 volts gives an output of −20 volts. Suppose, further, that a rule of design is that chains of logic may be used until an ON signal has deteriorated to −5 volts and an OFF signal has increased to −15 volts, then a restoration circuit must be used. Such a rule provides what may be called a two-to-one safety factor.

In some arrangements of circuitry the upper limit of the ON signal and the lower limit of the OFF signal may not be important. In other cases (for example, to control the back voltage on crystal diodes) it may be necessary to limit the voltage swings so that no signal is above 0 or below −20 volts.

Figure 11.6 illustrates the use of crystal diodes to "clamp" the signal level. In order for this to work successfully it is necessary that the −20-volt power supply be of very low impedance. Such low impedance may be obtained by having a heavy-duty supply or, preferably, by using regulation. Requirements of uniformity of performance with change of emission in the tubes used may lead to the use of clamping.

11.3.2. The Requirements of a D-C Level Restorer. From the discussion of the preceding section the ideal characteristics of a d-c level restorer may be established. For inputs greater than −10 volts the output is greater than 0, for inputs less than −10 volts the output is less than −20 volts, and if the input is −10 volts the output may be indeterminant.

In practice it is necessary to approximate the above characteristics, using tubes available from the manufacturer (some selection may be permissible); the characteristics should be substantially the same for aged tubes (whose emission has fallen off by, say, 50 per cent). These last requirements mean that there is an area in the vicinity of −10 volts where the performance of the circuit is indeterminant.

Vacuum tubes have the characteristic that the output is at a substantially higher d-c level than the input. For example, consider Fig. 11.7*a*, where the input is I and the output of the tube is O_1, which may quite possibly be 100 volts higher than I. The d-c level may be returned to the input level by use of a d-c coupling circuit (Fig. 11.7*b*). Such a coupling circuit requires a large negative voltage B in order not to give too much attenuation of the signal at O_1. In the following discussion it will be assumed that there is a loss of 50 per cent of the d-c signal at this attenuator. The high-frequency response is controlled by the capacitor C.

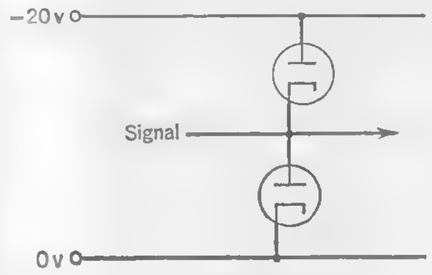


FIG. 11.6. Clamp circuit.

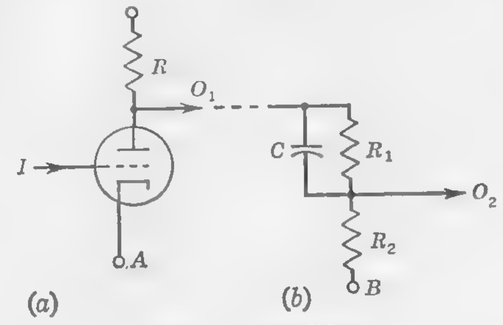


FIG. 11.7. Amplifier.

The disadvantage of the circuit of Fig. 11.7 is that the final output is at high impedance, and if the circuit is to drive several other circuits without loss in frequency response it will be necessary to use a cathode follower after O_2 (cf. Fig. 11.2). Another factor which may be a disadvantage in some circumstances is that the circuit of Fig. 11.7 inverts the signal.

11.4. BUFFER INVERTERS

The circuit of Fig. 11.7 is frequently called a buffer inverter, since there is signal inversion and the input is isolated from the output.

In the following discussion it is assumed that the tube is one-half of a 5687 twin triode, that the input I varies from −20 to 0 volt, that the output O_2 is clamped between −20 and ground by a circuit such as that of Fig. 11.6, and that the attenuation between O_1 and O_2 is 50 per cent.

Inspection of the plate characteristics of Fig. 10.15 shows that as I changes from −20 to ground the plate current changes from 0 to 40 mils. If the plate resistor R is 2.2K, then O_1 changes from 200 to 112 volts. If the tube has aged so that the emission has fallen to 50 per cent of its initial value, then O_1 swings only from 200 to 140 volts. Suppose that the resistors R_1 and R_2 (Fig. 11.7*b*) have been chosen so that a potential of 175 volts at O_1 produces a signal of −10 volts at O_2; under these circumstances the response of a new tube in the circuit of Fig. 11.7 is shown by the curve *ABCDEF* of Fig. 11.8 (cf. Fig. 11.1). If the emission of the tube is reduced by 50 per cent, the response curve changes to *ABGHEF*.

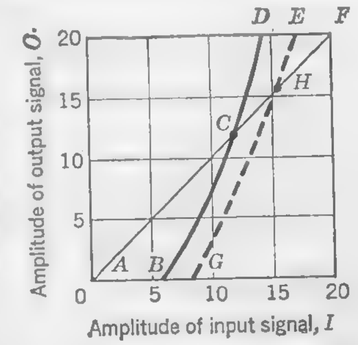


FIG. 11.8. Amplifier response.

With a new tube, unity gain occurs at point C or with an input amplitude of about 12 volts. With an old tube this point has moved to H, so signals less than 16 volts in amplitude will tend to die out in a chain of such circuits.

The response of Fig. 11.7 can be made more like the ideal (Fig. 11.1) by an "overdrive" technique illustrated in Fig. 11.9, in which the network R' and R'' has been chosen so as to raise the level of the input signal by about 5 volts. The condenser C' helps the high-frequency response.

For d-c applications, and this is particularly true of the 5687, it may not be possible to maintain the grid at near zero bias for substantial periods of time. In this case the

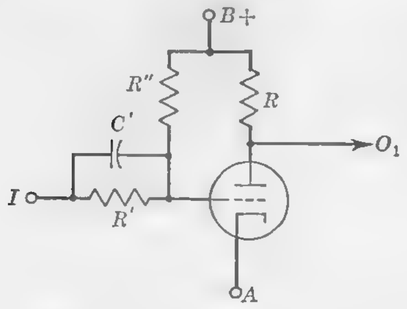


FIG. 11.9. Overdriven amplifier.

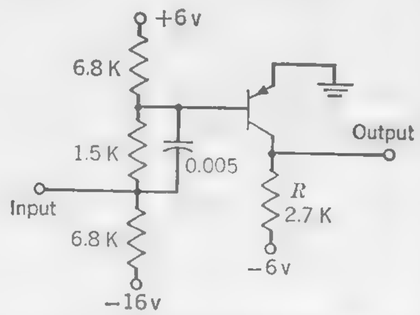


FIG. 11.10. An inverter circuit.

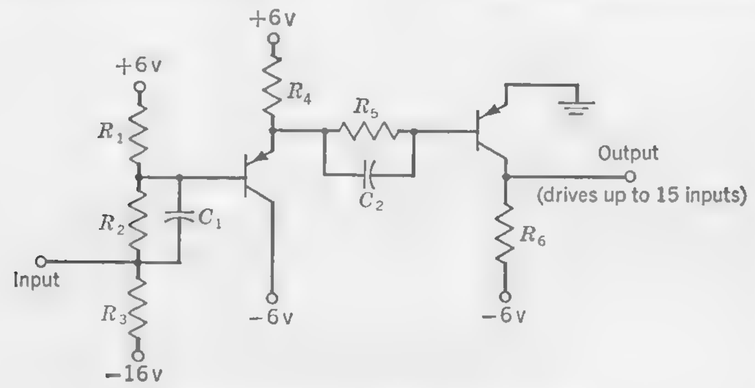


FIG. 11.11. An inverter-driver circuit.

overdrive technique is not practical. All the above discussion is valid for pulse signals, but for d-c signals the range of operation would have to be shifted. This shift in range may be accomplished by networks in the grid circuit or by putting current-limiting resistors in the cathode circuit.

For another approach to the problem of restoration of d-c levels see the section on Schmidt trigger circuits.

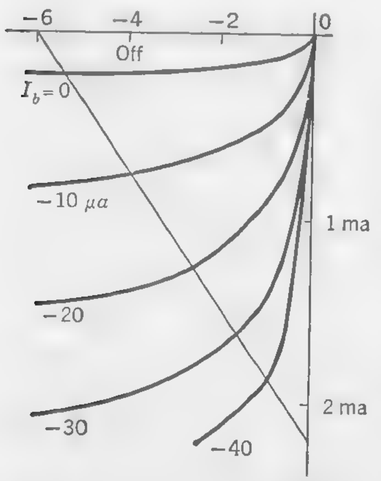


FIG. 11.12. Collector current-voltage characteristics.

11.4.1. A Transistor Buffer Inverter. A typical transistor inverter circuit is shown in Fig. 11.10. The load resistor R (2.7 kilohms) may not be present but may represent the output load. The transistor characteristics and the load line for the circuit shown may be as illustrated in Fig. 11.12. Assume that the input swings from 0 to −2.5 volts and that the current to the base of the transistor can be neglected (on the order of 0.05 ma), then the base of the transistor changes from about 1 volt to −2 volts. At plus 1 volt the transistor is cut off and the output is at −6 volts. With the base at −2 the transistor conducts and the output rises to about −0.1 volt. Even with an input of −0.2 volt the base will be 0.9 volt and the transistor will be safely cut off. The reverse bias of 0.9 volt on the base allows operation at junction temperatures above room temperature.

The condenser on the input network momentarily overdrives the transistor and improves the rise time of the output signal. This condenser should not be too large or the transistor will be driven into saturation and recovery will be slower.

A circuit such as that of Fig. 11.10 may be designed to drive as many as five other circuits of the same type. This means that the load on the collector may vary, depending upon the number of circuits driven. The lower level of the collector voltage may be fixed by adding a clamp diode to the appropriate potential. This improves the fall time of the circuit.

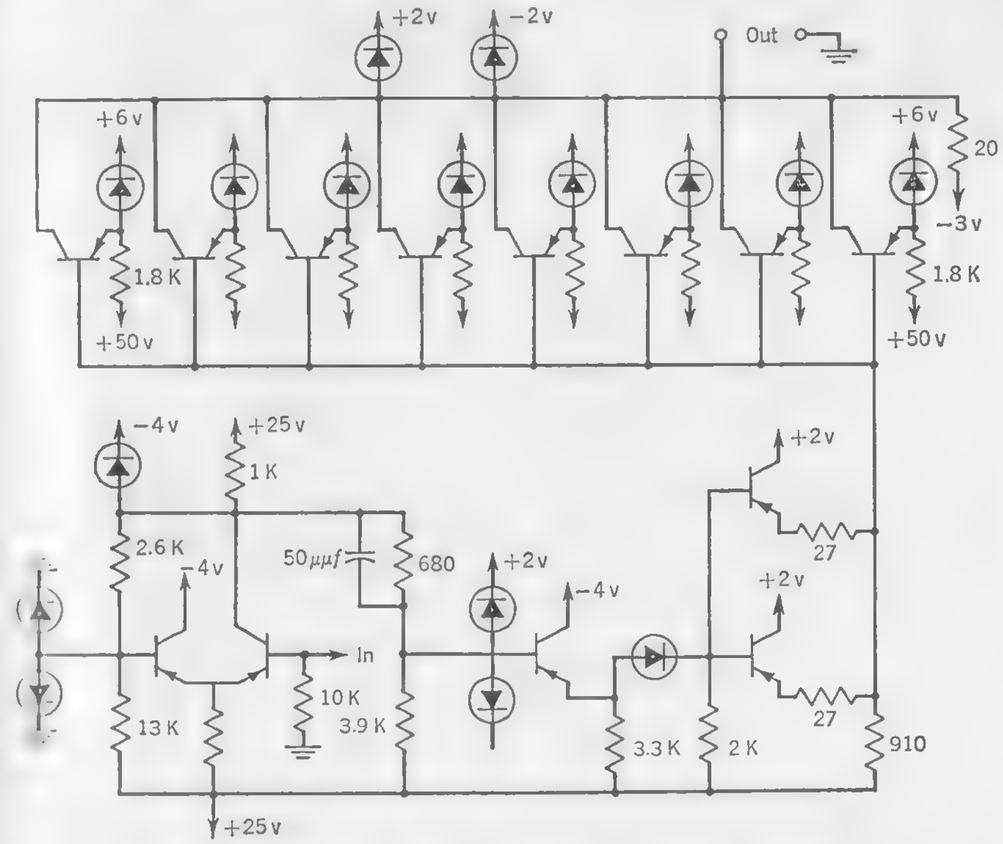


FIG. 11.13. 200-ma driver. (*Courtesy of University of Illinois.*)

If it is desired to drive more than five circuits the arrangement shown in Fig. 11.11 may be used. This can be designed so as to drive as many as 15 similar circuits.

High-current drivers may be designed by paralleling the output transistors. Figure 11.13 shows a 200-ma driver which will drive a load of 100 mmfd with rise and fall times of less than 20 mμsec (20×10^{-9} sec).

11.5. RELAY DRIVERS

Relay drivers are commercially available which drive 400-ma relays with an input signal of 50 μa.

The unit uses 4 ma at +12 volts and 20–30 ma at −12 volts, and depending upon the relay capabilities, can operate at 1 kc. Note that the output transistor must be protected against reverse surges by placing a clamping diode across the relay.

11.6. SIGNAL-MODIFYING CIRCUITS

Signal-modifying circuits accept a signal of one shape and put out a signal of a different shape. Amplifiers will not be considered here, although they may satisfy the

above definition. Generally, signal-modifying circuits play no role in the logic of the design (delay circuits may be an exception). Such circuits are required because of the difference between the input and the output signals of gating and amplifying circuits. For example, d-c levels must be restored, timing must be corrected, or safety factors in terms of levels or timing (or both) must be increased.

11.6.1. Clipping Circuits. In vacuum-tube circuitry it is necessary to prevent any noise occurring anywhere in the circuitry from being amplified to the point where it causes malfunctioning of the equipment. Since in most computer circuits signals are at two levels, high and low, it is only necessary to consider noise at these two levels. At the low levels vacuum tubes are operating below cutoff where the gain is very small. Thus, in most vacuum-tube circuitry, noise on low-level signals causes no difficulty. At high levels vacuum tubes are operating at zero bias and their gain is a maximum, so noise considerations become very important. With grids slightly above cathode potentials the effect of grid current (and the consequent lower impedance of the grid

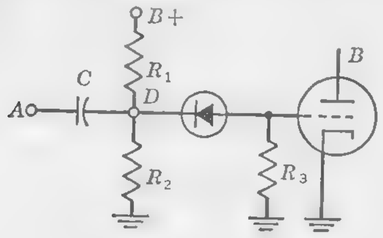


FIG. 11.14. Clipping circuits.

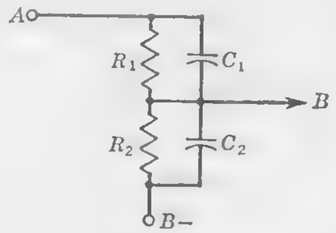


FIG. 11.15. Attenuation.

circuit) may tend to limit noise. If this is not enough, then a circuit such as that of Fig. 11.14 may be used. The bleeder resistors R_1 and R_2 establish a potential D at point D. Let XB denote the backward resistance of the crystal X (this may be 1 M), and let XF denote the forward resistance (100 ohms, say). If Eg is the potential at the grid, then for D greater than zero,

$$Eg = \frac{R_3}{R_3 + XB} \cdot D \text{ for } D > 0$$

and

$$Eg = \frac{R_3}{R_3 + XF} \cdot D \text{ for } D < 0$$

If $R_3 = 10$ kilohms, then $Eg = D/100$ for $D > 0$, and $Eg = 0.99D$ for $D < 0$.

11.6.2. Attenuation. In digital computer circuitry it is usually unnecessary to attenuate signals just to reduce their size. However, in vacuum-tube circuitry it is frequently necessary to change the d-c level. The most economical way to do this is to use a resistance divider, and the signal is attenuated whether one likes it or not. For example, in flip-flops it is necessary to d-c-couple the anodes back to the opposite grids (also see Art. 11.3.2).

A circuit which produces such a change in d-c level is illustrated in Fig. 11.15. Usually, linearity of such circuits is of no consequence. The d-c response must be correct and it should have good high-frequency response. If A and B denote the potentials at the respective points, then the d-c level at B is given by $B = A \cdot R_2/(R_1 + R_2)$. If the resistors R_1 and R_2 are sufficiently large the high-frequency response is given by $B = A \cdot C_1/(C_1 + C_2)$. Therefore, the least loss of signal occurs if R_2 is large and C_2 is small. R_2 being large (relatively) requires that $B-$ be a large negative potential. However, there is a practical limit to the potential $B-$ of a few hundred volts, and although the condenser C_2 may be left out altogether, the distributed capacity of the circuit supplies a lower limit. If C_1 is chosen relatively large, there will be considerable overshoot on the output signal at B. Usually this is of no importance except with respect to "recovery" of the circuit.

11.6.3. Ringing Circuits. Sometimes it is desired to generate a string of pulses from a square wave. This can be done with a ringing circuit such as that shown in

Fig. 11.16. The output here is actually a sine wave, but if it is sent to a tube biased below cutoff the output will be a train of pulses. Tube T_2 and the associated inductance are an oscillator. The tube T_1 switches the oscillator on and off. When the potential at A falls, the oscillator begins to oscillate, and when A rises again the current in tube T_1 damps out the oscillations. The variable resistor R provides gain control in the oscillator circuit.

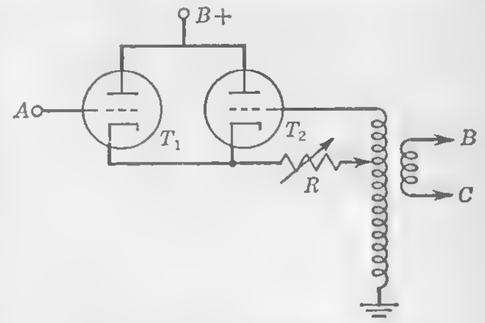


FIG. 11.16. Ringing circuit.

11.6.4. Pulse-train Generator. Figure 11.17 shows another way of generating a train of pulses. If a positive pulse is applied to point A this pulse passes down the delay lines L_0, L_2, L_3, and L_4. The pulse appears at the crystals at different times and there is an output at B for each one of these. The resistance R_2 is the characteristic resistance of the delay line to prevent reflections. The real design problem is to prevent reflections at the points where the crystals are connected to the delay lines.

Since there is attenuation in the delay lines, the signal at X_1 will be larger than that at X_5 (in a total of 8 μsec of delay this attenuation may be 2:1). Therefore, it is necessary that the large signal at X_1 does not see the low forward impedance of the grid when the grid is driven positive with respect to the cathode, and the signal at X_5 must still be large enough to drive the grid up to ground safely.

Note that the spacing between the pulses can vary here, whereas in the ringing circuit given above it had to be constant. Another important difference is that the delay-line circuit does not depend upon any continuous parameters, and there is relatively little restriction on the input signal. In the case of the ringing circuit the input

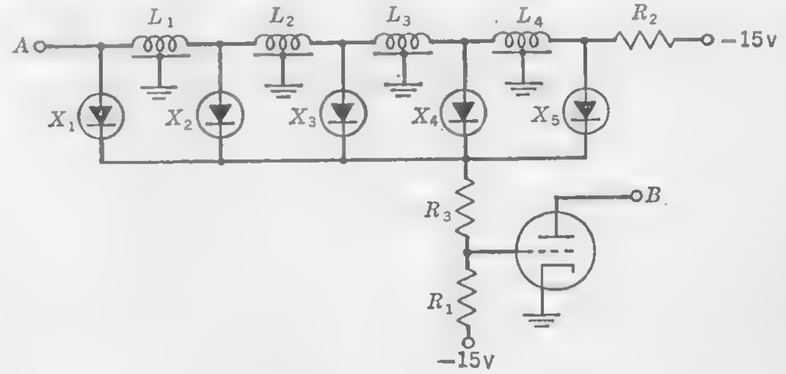


FIG. 11.17. Pulse-train generator.

signal must be of precise length, and at higher frequencies where the capacity in the oscillator-tank circuit is comparable to the circuit and tube capacities, the frequency may change when tubes are changed. Thus, for many applications in digital computers the delay-line circuit is to be preferred.

11.6.5. Delay Circuits. Circuits such as those shown in Figs. 11.18 and 11.19 can be used to delay pulses a fixed amount. In Fig. 11.18 a pulse at A causes the anode circuit to ring. The damping resistor R_2 must be of such size that the first positive swing will operate the following circuit but the second positive swing will not. Clearly it is very difficult to design such a circuit so that it will operate with a tube-emission change of 50 per cent. Thus, the circuit of Fig 11.19 is to be preferred. A positive pulse at A causes a negative pulse E to enter the delay line. This is reflected from the shorted end of the delay line and appears at C as a positive pulse F at a time t units later. With a 2:1 attenuation between signals E and F, delays of 8 to 10 μsec are easy to attain in this way.

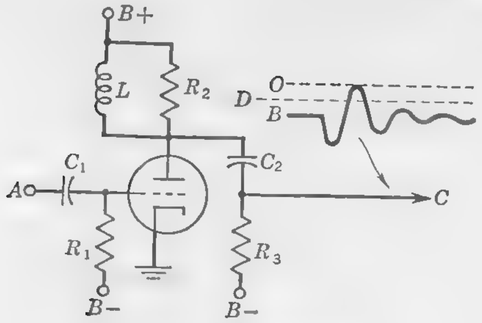


FIG. 11.18. Delay circuit I.

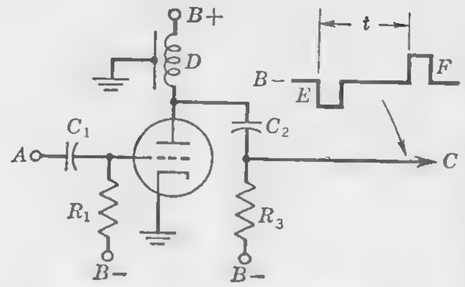
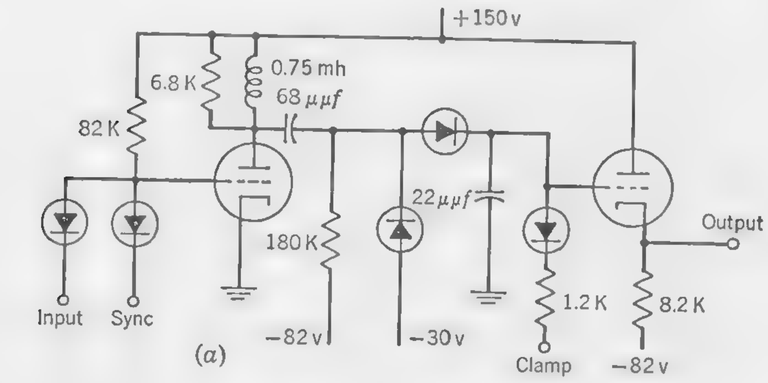


FIG. 11.19. Delay circuit II.



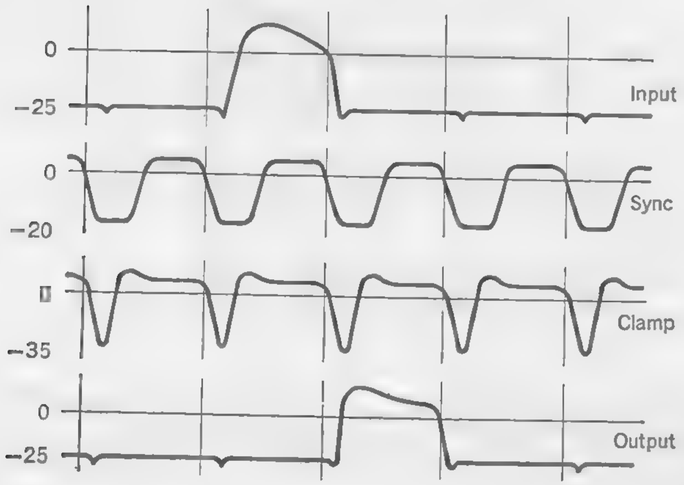
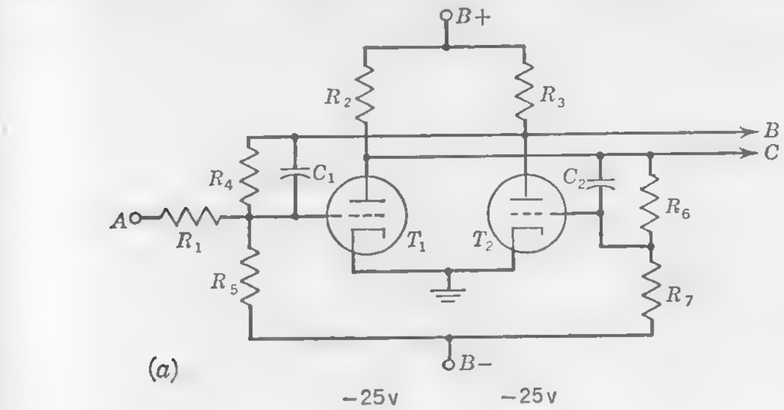


FIG. 11.20. (*a*) Delay circuit III. (*b*) Signals for delay circuit III.

Techniques such as that described in Art. 11.6.4 can be used to obtain two overlapping pulses at F and consequently to broaden the output pulse.

The author's first contact with the delay circuit of Fig. 11.18 was at Cambridge University on the EDSAC project in 1947. More recently, a version of this circuit, shown in Fig. 11.20*a*, has been used in this country [4, p. 416]. The waveforms of the input and output signals are shown in Fig. 11.20*b*.

11.6.6. Schmidt Triggers. In connecting electronic equipment with mechanical equipment it is often necessary to use long time constants in the circuits to eliminate the effects of contact bounce or chatter. On the other hand, steep signal fronts may be needed to operate pulse circuitry properly. The Schmidt trigger serves such a purpose. If a slowly rising signal is applied to point A in Fig. 11.21*a* nothing happens to outputs B and C until a certain potential is reached. At this point tube T_1 begins to conduct, and the positive feedback in the flip-flop-type circuit causes tube T_2 to be cut



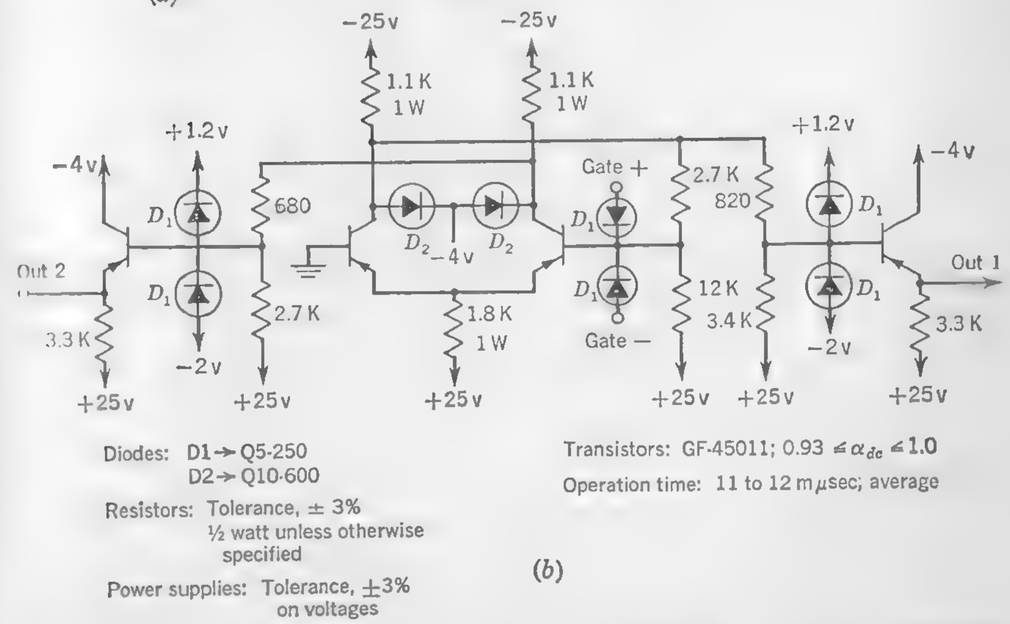


FIG. 11.21. (*a*) Schmidt trigger. (*b*) Transistor Schmidt trigger. (*Courtesy of University of Illinois.*)

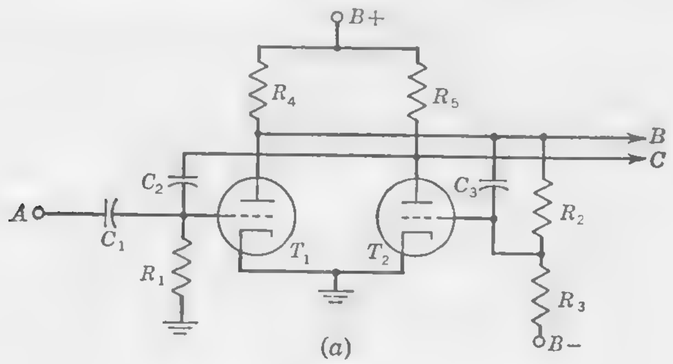
off. As the potential at A lowers there is again a point (not necessarily the same) at which the flipping action takes place. Thus the output levels of B and C are determined by the d-c level at A, and any transition takes place at rates independent of the input signal (the transition depends only upon the frequency response of the feedback loop). Note that B and C are complementary in nature, that is, if B is high C is low, and vice versa.

Figure 11.21*b* shows a high-speed transistor Schmidt trigger. The circuit provides complementary outputs and introduces a delay of only 11 to 12 millimicroseconds.

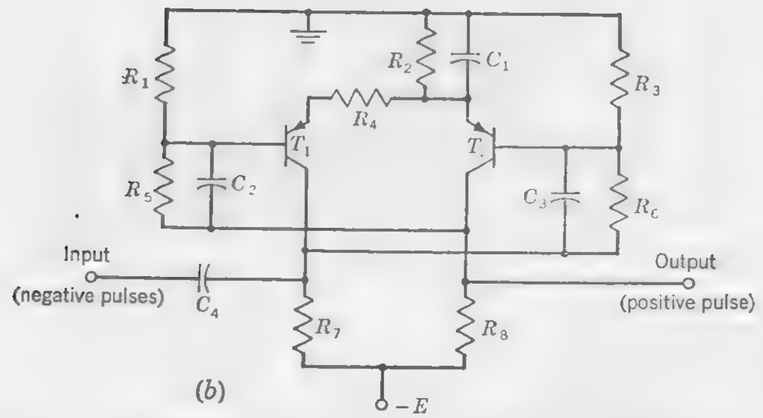
11.6.7. Pulse-broadening Circuits. One-shot multivibrators (Fig. 11.22*a* and *b*) may be used to transform a pulse into a gate signal. The length of the resulting signal depends only upon the internal time constants of the circuit and may range from a

few microseconds up to seconds. Since the delay time depends upon the internal time constants this is not a good way to get precise delays. Delays of short duration (a few microseconds) can be obtained with the circuit of Fig. 11.22*b*.

The operation of the one-shot multivibrator is as follows. The normal state is with tube T_1 conducting, since the resistor R_1 returns the grid to ground. A negative pulse at A turns this tube off, and the positive feedback causes a flip-flop action driving the grid of T_1 even more negative. The condenser C_2 (and C_1) recharges with time constants, depending upon R_1 and R_5, and the moment that T_1 begins to conduct the circuit flips back to its initial state. Thus the length of the signal depends upon the condensers C_1 and C_2 and upon the resistors R_1 and R_5.



(*a*)



(*b*)

FIG. 11.22. (*a*) One-shot multivibrator. (*b*) Transistor one-shot multivibrator.

Figure 11.22*b* shows an example of a transistor one-shot multivibrator. A negative pulse on the input will cut off transistor T_2. The rise in potential at the collector will cause T_1 to conduct. The circuit will remain in this state until the circuits recharge, at which time the circuit will "flip" back. The unbalance required to make a one-shot circuit may be inserted into the basic multivibrator circuit in many ways (resistor R_4).

The circuit of Fig. 11.23 gives a more precise delay (pulse width), and operates in the range from a few tenths of a microsecond to several microseconds (depending upon the attenuation of the delay line).

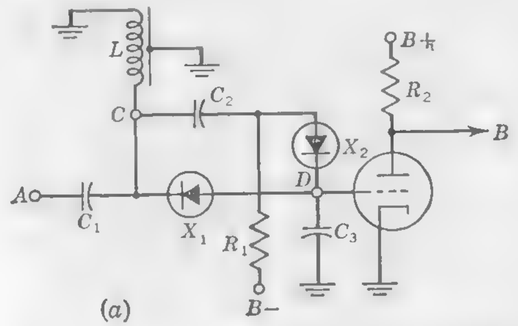
Assume that the tube of Fig. 11.23*a* is conducting. A negative pulse at A, via diode X_1, charges C_3 negatively and cuts the tube off. This negative pulse passes down the delay line L, is reflected, and appears at C sometime later as a positive pulse. This positive pulse causes diode X_2 to conduct and charge the condenser C_3 positively The waveforms are shown in Fig. 11.23*b* for a delay line of length $t/2$.

Note that the forward currents through the two diodes must be able to charge or

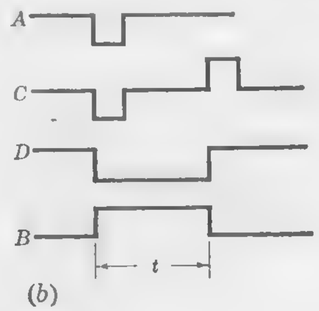
discharge condenser C_3, whereas the back currents must change its charge very little. In practice vacuum diodes work best, and then it may be necessary to add a very large resistor from the grid of the triode to ground so that it will remain in the conductive state in the absence of input pulses.

11.6.8. Pulse-generating Circuits. The circuits discussed in Art. 11.6.7 transform pulses into gate signals (i.e., a pulse of much longer duration). This article considers circuits which accomplish the complementary function, which is that the circuits transform gate signals (long pulses) into pulses (short pulses).

When current flows in the tube of Fig. 11.24, the low forward resistance of the diode X prevents any oscillation. Whenever the potential at A goes negative (sufficiently fast), the tuned circuit consisting of L and the associated capacity begins to

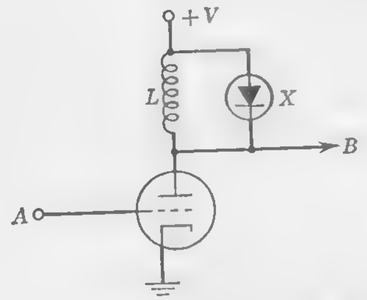


(*a*)



(*b*)

FIG. 11.23. Pulse-broadening circuit.



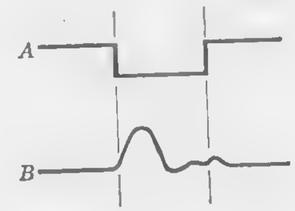


FIG. 11.24. End-pulse circuit.

ring. However, on the second half of the first cycle as the potential at B falls below the power supply voltage V the low forward resistance of X quickly damps out the oscillation. Thus, the output is one-half cycle of a sine wave at the resonant frequency of the anode circuit. When A swings positive there is almost no output signal due to the low forward resistance of the diode X.

Note that the input signal at A must fall fast enough or the tube acts as a dampening resistance to the tuned circuit. Such a circuit is operating at zero bias and is extremely sensitive to noise, so it may be necessary to include the network of Fig. 11.14 in the grid circuit.

11.7. AMPLIFIERS*

11.7.1. Introduction. In an amplifier, energy obtained from a local source is directed to a load under the control of an input signal. In most cases input variations will be supposed to cause simultaneous variations at the load, but a certain amount of delay between input and output may be present. This delay may be either an essential characteristic of the amplifier mechanism, it may be intentionally introduced by

* By Bram J. Loopstra.

the choice of circuit parameters, or it may be just an undesirable effect that should be minimized in order to obtain best results.

The purpose of electronic amplifiers will usually be to increase the amplitude of the voltage, current, or power at the output with respect to the input. The local energy source is often called an "active" element, and in the next part of this section it will be assumed that it is realized by means of an electronic vacuum tube. Other possible active elements, such as transistors, will be considered elsewhere.

It should be noted that the definition of an amplifier given here means that to be an amplifier the device must be able in principle to give power amplification (though this may not always be actually obtained). Passive elements such as transformers can give voltage or current amplification but no power gain, and are therefore not amplifiers in the strict sense.

The output impedance of an amplifier may be made to have almost any desired value independent of the impedance of the input. This gives rise to an extensive use of some types of amplifiers as impedance-matching elements. In many cases where pulse-type input signals must be handled, the purpose of the amplifier may be to amplify only one range of frequencies contained in the input signal so as to modify its shape; this kind of shaping amplifier will be treated in the next article. The most important characteristics of an amplifier are its (complex) gain (voltage, current, or power) and the range of frequencies over which substantially constant gain may be expected (the bandwidth). Bandwidth can be defined as the range of frequency between the two points on the frequency scale where the gain is some predetermined fraction of the gain in the middle of this range. This definition holds if a lower and upper limit are really present; in the case of d-c amplifiers the lower limit is always zero frequency.

Amplifiers can be classified according to the position of their passband on the frequency scale.

1. Direct-coupled amplifiers. Passband from direct current to an upper limit that may be anywhere between 5 kc and 5 mc.
2. Audio amplifiers. The frequency ranges from 20 cycles to 20 kc or less.
3. Video amplifiers. Wide-band amplifiers with a frequency range from below 100 kc to 5–20 mc.
4. Tuned amplifiers.
   a. Narrow band
   b. Wide band (bandpass amplifiers)

The terms wide and narrow with respect to frequency bands have a relative rather than an absolute meaning. If f_m denotes the center frequency of the band and f_l and

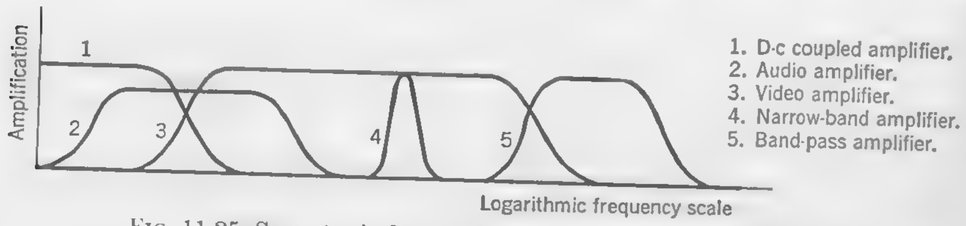


FIG. 11.25. Some typical examples of frequency-response curves.

f_u the lower and upper frequency limits respectively, then if $(f_u - f_l)/f_m \ll 1$ the band is called narrow. Thus an amplifier with a range from 1–50 kc is a wide-band amplifier; whereas with a range from 1,001–1,050 kc a narrow-band amplifier is obtained, though the bandwidth is the same in both cases. Figure 11.25 gives an impression of the form of the frequency characteristics in the different cases.

A second classification of amplifiers can be given according to the purpose for which they are used. Thus, there are voltage, current, power, and shaping amplifiers. In an amplifier consisting of several stages the function of different stages may not be the same; the arrangement where several stages of voltage amplification are followed by a power-amplifying stage is a typical example.

Finally, a classification can be given based on the adjustment of the tubes in the amplifier.

1. Class A amplifiers. Here the grid bias and the amplitude of the input signal have been so chosen that the anode current will never be zero during the electrical cycle.
2. Class AB. The anode current is zero during a small part of the cycle.
3. Class B. The grid bias is about equal to the cutoff value. In the absence of input signals the anode current is approximately zero. If a signal voltage is applied current will flow during the positive half of the cycle.
4. Class C. The grid bias is substantially larger than the cutoff value. Current flows during part of the positive half of the cycle if the amplitude of the input signal is sufficiently large.

A suffix 1 or 2 is sometimes added to the letters A, B, and C to distinguish the case where grid current does not flow during any part of the cycle (suffix 1), from the case where grid current does flow because the input amplitude is such as to drive the grid positive with respect to the cathode during part of the cycle (suffix 2). This last classification applies to amplifiers handling input signals that have a periodic (usually sinusoidal) character. It is little used for pulse-amplifier applications.

Before discussing the properties of amplifiers of different kinds, it will be necessary to give some account of the different categories of signal which they may be called upon to handle, as the amplifier specification will be largely dependent on the essential features of the input signals. A signal is useful only to the extent that it carries information; the elements of the signal representing the "message" must be faithfully reproduced by any amplifier used to handle it, whereas other features of the signal may be allowed to get lost or suffer distortion. A signal can contain information in many ways. The simplest method is to use a d-c signal and to consider the amplitude of voltage or current to be representative of the message. If an a-c (sinusoidal) signal is used, amplitude, phase, and frequency can be varied to make up the information. Pulse trains can be used and the information given in the form of pulse width, pulse distance, etc.

If any of the types of signals mentioned here is used to modify a sinusoidal signal (usually of a much higher frequency) the process is called modulation, and the sinusoidal signal is the carrier. In the case where such quantities as voltage, frequency, time intervals, etc., are continuously variable, the information is called analog. If on the other hand these quantities have only a limited number of possible values (the absolute values are not usually of interest) it is a coded or digital system.

Without aiming at completeness one can distinguish the following types of signals:

1. Pure sine waves and other periodic signals
2. Modulated sine waves
3. D-C signals
4. Pulse signals (of a nonperiodic character)

The difference between the categories 3 and 4 is especially important in computer applications. Type 3 signals can have two values, both of which can last for an indefinite period of time. This means that they cannot be handled by amplifiers where the d-c level will go back to some no-signal value after a sufficient lapse of time; this is the case in all a-c-coupled amplifiers. In type 4 signals the signal can also have two levels, but only one of these can last for a long time. Therefore, if some precautions are taken, these signals can be handled by a-c-coupled amplifiers. Figure 11.26 shows some examples illustrating these cases.

FIG. 11.26. Examples illustrating the difference between d-c and pulse-type signals.

11.7.2. Distortion. In a perfect amplifier the output signal would be exactly similar to the input signal, the only difference being a different amplitude of voltage, current, or both. In actual amplifiers this relation does not hold, and this phenomenon is generally called distortion. Its effects may be objectionable or not, as the case may be. Any waveform applied to the input can be resolved, by means of a Fourier

analysis, into a finite or infinite number of purely sinusoidal components with proper relative amplitudes. It will be clear, therefore, that undistorted output can be expected only if

1. All frequencies present in the input signal are also present in the output signal with the same relative amplitude
2. The output contains no frequencies not present in the input signal
3. The phase shift is either zero, or 180 deg, or proportional to frequency

The different kinds of distortion resulting from the fact that none of these conditions is satisfied in an actual computer are the following:

1. Nonlinear distortion, also called amplitude distortion. This is caused by the curvature of the dynamic tube characteristics. As a result condition 2 is not satisfied, and harmonic and intermodulation frequencies will be present in the output. Extreme cases of nonlinear distortion result when the amplitude of the input signal is such that the normal (approximately linear) part of the transfer characteristic is exceeded. Figure 11.27 gives an example of this situation.

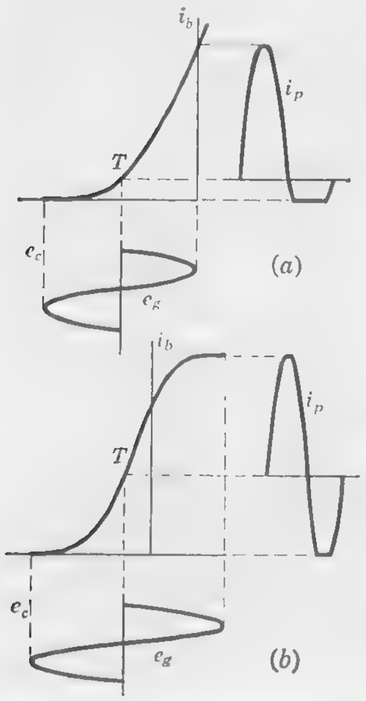


FIG. 11.27. Nonlinear distortion.

2. Frequency distortion is caused by the variation of gain with frequency beyond the limits of the passband. It is the result of the frequency dependence of tubes (interelectrode capacitances, lead inductances) and associated circuitry (bypass and coupling condensers, wiring capacitance). The relative frequency amplitudes in the output will differ from those at the input and condition 1 is violated.

3. Phase distortion has the same causes as frequency distortion and results in a phase shift that is not in accordance with condition 3.

Frequency and phase distortion will often be negligible for those frequencies which lie in the passband of the amplifier.

11.7.3. Noise. A fundamental limit to the amount of amplification that can be used in an amplifier is set by the presence of noise. Noise consists of random voltage and current variations with frequency components extending over a wide range of frequencies. They originate both in the signal source and in the tubes and components of the amplifier. Any input signals smaller than the input-noise voltages cannot be successfully amplified. Signals whose amplitude is somewhat in excess of the noise can be amplified and detected in the output with a certain probability of error. In systems having no redundancy in their coding and using no error-correcting or error-detecting codes, they are still useless, since uncertainty about the information, even if very small, cannot be tolerated. The signals used in computer applications will usually satisfy the condition that they be much larger than the noise present so that they can be amplified with negligible risk of being masked by noise signals in the output.

Thermal agitation noise is the result of the motion of free electrons in a conductor. Its magnitude can be expressed as

$$E^2 = 4kTRf \tag{11.8}$$

where E = effective value of noise voltage
k = Boltzmann's constant = 1.374×10^{-23} joule/°K
R = resistive component of the impedance across which the noise is measured
f = the frequency range considered

In some resistors the noise level is found to be much higher than it should be according to Eq. (11.8). This is especially true for solid-carbon resistors, and the reason is that these have a granular character; the variation in contact resistance between the particles gives rise to noise in excess of the thermal agitation noise. This type of resistor should therefore not be used in the signal leads of low-level amplifier stages. Wire-wound resistors are probably best for these applications.

Tube noise can be of several varieties. The best known of these is shot noise caused by small variations in the emission of the cathode which give rise to anode-current variations. Another important noise effect in multielectrode tubes is partition noise. If the cathode current of a pentode could be held absolutely constant the fraction going to the screen would still show small variations, and noise would appear in the output. In triodes this effect is absent.

The phenomena called static and atmospherics are sometimes also considered as noise sources. They cause troubles only if the amplifier is designed to receive signals transmitted through space. In other applications good shielding is a means of eliminating these spurious disturbances originating from thunderstorms, etc.

A cause of trouble that is somewhat related to noise is interference of different kinds, such as hum, microphonic effect, and pickup from other signal leads. In these cases the disturbances have not a random but a well-defined character and can be reduced to sufficiently safe values, or even completely eliminated by proper effort. Hum is the result of the introduction into the circuit of an amplifier of power-line frequency voltages, or currents by tube heaters, insufficiently filtered power supplies, or stray magnetic and electrostatic fields from power transformers. If the amplifier does not discriminate against low frequencies even small amounts of hum in low-level stages will be very troublesome, since the hum voltage will be amplified as well as the signal voltage. If low-frequency response is not needed, it is therefore advisable to attenuate these frequencies as much as possible. Reduction of low-frequency response is also a very good method to deal with microphonic effects. This effect results from the variations of the characteristics in tubes and components under the influence of mechanical vibrations and shocks. The frequency components of the disturbing signals usually lie in the low-frequency range, and if this is outside the passband of the amplifier little trouble will be experienced. In other cases shock-mounting is the only remedy. Pickup of signals from insufficiently shielded leads in the vicinity of an amplifier can only be avoided by removing the leads to a safe distance or by better shielding. Interference of this kind often occurs when ground connections are not made the right way and coupling between circuits via the ground (chassis) is introduced.

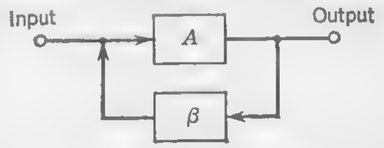


FIG. 11.28. Block diagram of a feedback amplifier.

11.7.4. Feedback. Both distortion and noise can be partly eliminated, at the cost of gain, by the use of negative or inverse feedback. In a negative-feedback amplifier, part of the output is fed back to the input so that it subtracts from the input signal; Fig. 11.28 shows the block diagram for this arrangement.

If A is the gain of the amplifier without feedback and β the gain of the feedback loop, 1β is called the feedback factor. The gain with feedback is

$$G_f = \frac{A}{1 - A\beta} \tag{11.9}$$

Normally the feedback factor is much larger than unity and $G_f = 1/\beta$. This means that the gain is now only dependent on the characteristics of the feedback circuit which will probably consist only of passive elements. The sensitivity of the amplifier to changes in the A circuit is reduced by a factor $1/(1 - A\beta)$.

Equation (11.9) also means that the gain will be independent of the load, if the load is not part of the β circuit. Frequency and phase distortion are both dependent on tube characteristics and load impedance, and the use of negative feedback will give improvement in this respect. Furthermore, it can be shown that nonlinear distortion is reduced by the factor $1/(1 - A\beta)$, compared with an amplifier of the same gain without feedback.

To understand what the function of negative feedback can be in the reduction of noise we refer to Fig. 11.29. The A circuit has been split into two parts, with gain A_1, and A_2, and a noise generator is situated between the two parts; it can be shown that for a noise source of this type the feedback improves the signal-to-noise ratio by a factor $1 - A\beta$. It will be clear, however, that no improvement can be obtained in the signal-to-noise ratio due to noise at the input of the amplifier. Feedback is useful, therefore, to reduce the effects of badly filtered power supplies, but does not help very much against the thermal agitation and tube-noise effects in low-level amplifier stages.

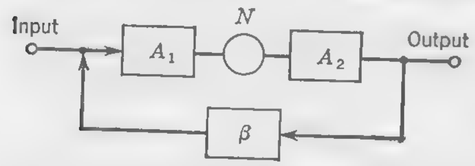


FIG. 11.29. Feedback amplifier with noise source.

11.7.5. Direct-coupled Amplifiers. The problems associated with d-c-coupled amplifiers depend very much on the type of signals they are expected to handle. The most difficult case is the d-c amplification of small signals (low-level case), and this will be treated first.

The difficulties of small-signal amplification arise from the fact that in such an amplifier there are no means of distinguishing between the desired signal and the slow variations in tube and circuit characteristics and supply voltages, usually called "drift," which also causes changes in the d-c levels in the circuit. For this reason some kind of zero adjustment will be necessary in this kind of amplifier.

Another problem is connected with the difference between d-c plate and grid potentials; in most cases the use of batteries to overcome this situation is not attractive, and voltage dividers, with their inherent loss in gain, have to be used.

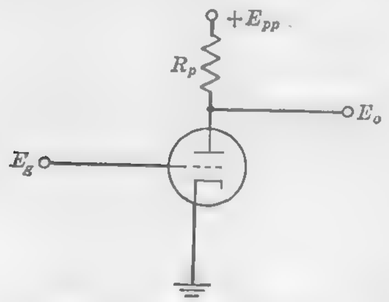


FIG. 11.30. D-C-coupled triode amplifier.

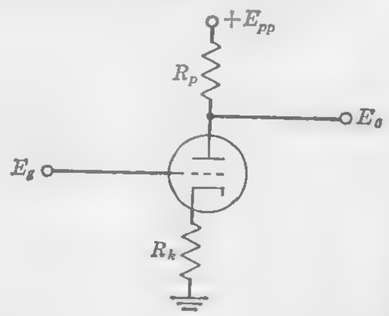


FIG. 11.31. D-C-coupled triode amplifier with a cathode resistor.

Referring to Fig. 11.30, there are the following formulas for the plate current and the output voltage:

$$i_p = \frac{Epp + \mu E_g}{r_p + R_p}$$

$$E_0 = \left(\frac{E_{pp} r_p / R_p - \mu E_g}{1 + r_p/R_p}\right)$$

where μ is the amplification factor and r_p the plate resistance of the tube. The voltage gain is

$$G = \frac{\partial E_0}{\partial E_g} = \frac{-\mu}{1 + r_p/R_p}$$

Therefore, voltage gain $\rightarrow \mu$ if $R_p \gg r_p$.

In using these formulas it should not be forgotten, however, that r_p is approximately inversely proportional to plate current for small values of i_p and that μ decreases with reduction of i_p. Beyond a certain limit the amplification can therefore not be increased by using a higher value of R_p. To obtain more gain in that case, a higher value of Epp should be used.

The introduction of a cathode resistor into the circuit of Fig. 11.30 gives the arrangement shown in Fig. 11.31. The effect of the cathode resistor is to introduce negative feedback and the gain is therefore reduced. The corresponding formulas are

$$i_p = \frac{(Epp + \mu E_g)}{[r_p + R_p + (\mu + 1)R_k]}$$

$$Eo = \frac{[r_p + (\mu + 1)R_k]\, E_{pp}/R_p - \mu E_g}{1 + [r_p + (\mu + 1)R_k]/R_p}$$

$$G = \frac{-\mu}{1 + \dfrac{r_p + (\mu + 1)R_k}{R_p}}$$

The output impedance (plate) is

$$Z_p = \frac{r_p + (\mu + 1)R_k}{R_p + r_p + (\mu + 1)R_k} R_p$$

If μ is large and $\mu R_k \gg r_p + R_p$, the formula for gain reduces to

$$G = \frac{R_p}{R_k}$$

This illustrates the fact that a large amount of negative feedback can make the gain practically independent of tube characteristics.

If a pentode tube is used instead of a triode and fixed screen potential is assumed, the circuit of Fig. 11.32 is obtained. Since the μ of pentodes is not a very well-defined quantity, it is better to use the transconductance g_m in the formulas. The gain is now found to be

$$G = \frac{-g_m R_p}{1 + R_p/r_p}$$

r_p in a pentode is normally very large, so $G = -g_m R_p$. In general, g_m is roughly proportional to i_p for sharp cutoff pentodes, and the gain is therefore proportional to the voltage drop across the anode load register R_p.

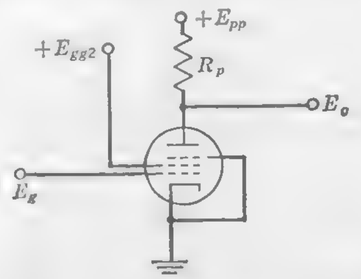


FIG. 11.32. D-C-coupled pentode amplifier.

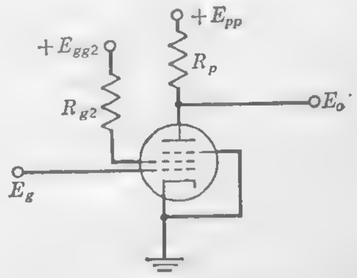


FIG. 11.33. D-C-coupled pentode amplifier with screen-grid resistor.

When a screen-grid resistor is added, the circuit of Fig. 11.33 is obtained. The analysis of this circuit is based on the assumption that the cathode current divides between screen grid and anode in the substantially constant ratio $\alpha = ig_2/i_p$ and that both i_{g2} and i_p are independent of plate voltage. Then the gain can be written as

$$G = \frac{-R_p g_m}{1 + \alpha R_{g2} \partial i_p / \partial E_{g2}}$$

Coupling between successive stages in a d-c-coupled amplifier is a difficult problem because the d-c level at the anode of one stage and at the grid of the next stage are usually widely different. When a number of voltage-supply levels is present, the

arrangement of Fig. 11.34 can sometimes be used. If only one supply voltage is available the second stage can be provided with a rather large cathode resistor; this will reduce the gain but will give a much more stable arrangement. This method of coupling should always be used, if circumstances permit, in the first stages of a low-level amplifier. At higher levels the drift introduced by coupling elements is less serious.

The best way of coupling (in theory) would be to use batteries (Fig. 11.35). However, these would have to be floating, with resulting high cost and bulk. Also, poor high-frequency response may be expected, since the ground capacitance of this coupling element is high. This method, therefore, is not much used.

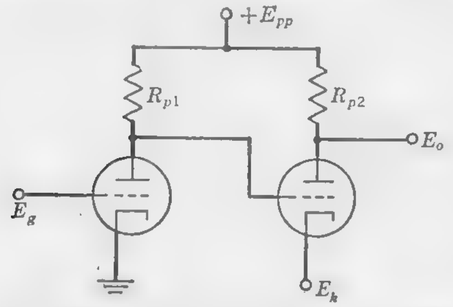


FIG. 11.34. Coupling of two d-c amplifier stages if a separate cathode supply voltage is available.

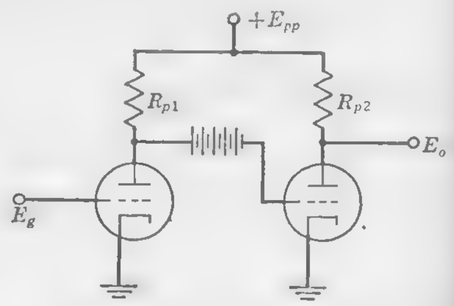


FIG. 11.35. D-C coupling between stages using batteries.

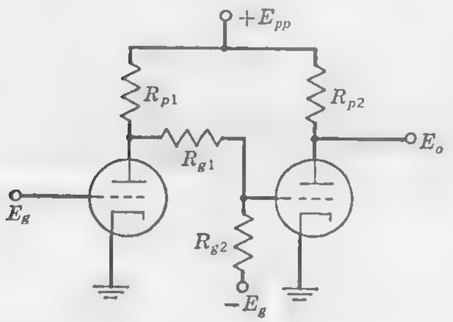


FIG. 11.36. D-C coupling between stages using resistors.

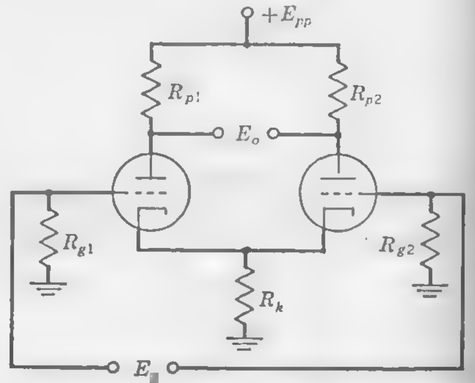


FIG. 11.37. Balanced arrangement used to eliminate power-supply variations.

A common method of coupling employs a voltage divider between the anode and the grid of the next stage (Fig. 11.36). This will always cause a loss of gain, but in cases where a negative supply voltage is available, it can be kept to a value below 3–6 db.

The zero drift in d-c-coupled amplifiers is mainly caused by three factors: tube aging, heater-voltage variations, and power-supply variations. The use of similar tubes that show the same variations as the amplifying tubes and can give a canceling effect is the best way of obtaining improvements.

Elimination of power-supply variations can be done by using some kind of differential arrangement, such as the one shown in Fig. 11.37. If absolutely identical tubes and heater voltages are assumed and the input signal is applied to the tubes in opposite polarity, the output between the two anodes will be independent of power-supply variations, since these have the same polarity on both sides. Negative resistive feedback can be very useful to obtain high linearity and good stability of gain, but it is of relatively little use in overcoming zero-drift troubles.

The effects of shunt capacitances in the circuits discussed have not been considered because this type of amplifier is not usually expected to handle high frequencies. The very high reactance of the shunt capacitance may, therefore, be assumed to have negligible influence on the operation of the amplifier. D-C-coupled low-level amplifiers are relatively infrequently used in computer applications. They are found in vacuum-tube voltmeters, regulator equipment, servo amplifiers, etc.

High-level d-c amplifiers, on the other hand, are widely used in computers. They are necessary for the manipulation of d-c signals, which cannot be handled by a-c-coupled amplifiers. The formulas for the low-level case still apply in principle when the input signals of the amplifier have values in the order of 1–10 volts or more, but in most cases the tube characteristics (r_p, g_m, μ) occurring in them can no longer be treated as constants. They are, therefore, of relatively little use in determining the exact behavior of the circuit under these conditions, and normal practice is to refer to the manufacturer's data and curves to obtain the desired information.

Voltage-divider coupling between stages is generally used, as the loss in gain is not serious in most cases. Nevertheless, the use of 1 or 2 per cent high-stability resistors in the divider networks may be advisable to keep margins as large as possible. The troubles encountered as a result of drift in the low-level case are practically absent here. Tubes are very often either cut off or drawing a current that must satisfy the condition of being in excess of some specified value to obtain correct operation of the circuit. On the other hand high-frequency response is often of great importance.

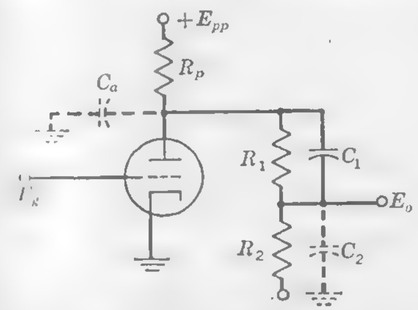


FIG. 11.38. D-C-coupled amplifier with speeding-up condenser for improved high-frequency response.

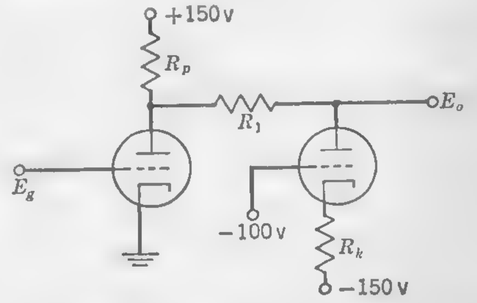


FIG. 11.39. The use of a current source to obtain a constant voltage drop.

Figure 11.38 shows how a capacitor across the upper resistor of the voltage divider can be used to compensate for the effect of the input capacitance of the next stage. To obtain exact compensation, the reactances of the capacitors C_1 and C_2 should be in the same ratio as the values of the resistors R_1 and R_2. The effect of the anode capacitance remains uncompensated; improvement can be obtained by using a peaking inductance in series with Rp. This method is discussed in more detail in the section on RC-coupled amplifiers.

In cases where the loss in gain resulting from the use of a voltage divider would still be objectionable, the upper resistor may be replaced by a neon glow tube (shunted by a capacitor). Excessive noise introduced by the glow tube limits this possibility to the high-level case.

Another solution that gives the same result is shown in Fig. 11.39. Here the resistor R_p of Fig. 11.38 has been replaced by the triode circuit, which acts essentially as a current source, maintaining a constant voltage across R_1.

It is interesting to note that two such circuits as the one in Fig. 11.38 with inputs and outputs cross-connected constitute the circuit of the well-known Eccles-Jordan trigger. For this reason this amplifier has been used in some large-scale computers as the element from which all necessary amplifiers and memory circuits were constructed (cathode-followers being used to give a low-impedance output). Figure 11.40 gives a practical example of this type of circuit.

The cathode follower (Fig. 11.11) is an amplifier with a relatively large resistor in the cathode lead and output taken from the cathode. Figure 11.41 also shows the shunt capacitances which influence the high frequency response of this amplifier.

Figure 11.42 shows the equivalent circuit on which the analysis can be based; z_b is the total load impedance, including the cathode resistor (if present). It is assumed that one side of the heater is grounded, so C_{pk} and C_{hk} may be considered to be in

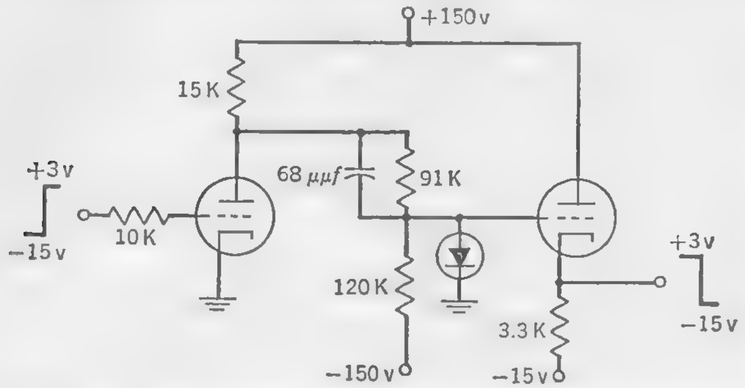


FIG. 11.40. Practical example of a high-level d-c-coupled amplifier.

parallel. The gain at frequency $f = \omega/2\pi$ is given by

$$G = \frac{(j\omega C_{gk} r_p + \mu) z_b}{j\omega(C_{gk} + C_{pk} + C_{hk}) r_p z_b + r_p + (\mu + 1) z_b}$$

At sufficiently low frequencies this reduces to

$$G = \frac{\mu z_b}{r_p + (\mu + 1) z_b}$$

Apparently G will always be smaller than unity. If $z_b \gg r_p$, $G \to \mu/(\mu + 1)$. In practice, values > 0.9 are obtained with loads in the range from 500 ohms–10 kilohms.

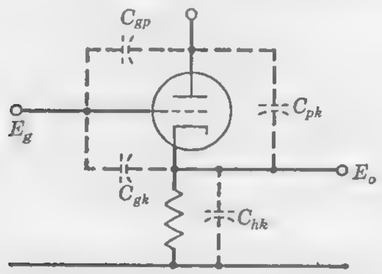


FIG. 11.41. Cathode-follower circuit showing interelectrode capacitances.

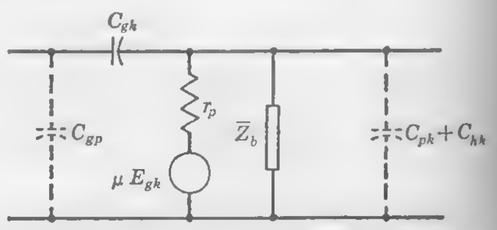


FIG. 11.42. Equivalent circuit of a cathode follower.

The input capacitance is found to be $C_i = C_{gp} + (1 - G)C_{gk}$, or in practical cases, $C_i = C_{gp} + 0.1C_{gk}$.

If C_{gp} is small (pentode tube), this means that very low values of input capacitance can be obtained. The output admittance consists of several terms:

1. The admittance of z_b
2. The admittance of $C_{pk} + C_{hk}$
3. The admittance of C_{gk} in series with C_{gp} in parallel with the internal impedance of the source
4. The admittance of the tube

The tube admittance can be found by calculating the current that flows as a result of an alternating voltage V impressed on the output terminals $Y = V + \mu E_{gk}/r_p V$, which reduces to $Y = 1 + \mu/r_p$ as the gain is approximately unity. For large values

of μ this may be further reduced to $Y = g_m$. If the other terms in the output admittance can be neglected we have output impedance

$$Z_0 = \frac{1}{g_m}$$

In practical cases this may be as low as a few hundred ohms. Because of its low input capacitance and output impedance the cathode-follower amplifier is widely used for power amplification and impedance matching. Practically flat frequency response can be obtained from direct current to about 1 mc. Because of the inherent negative

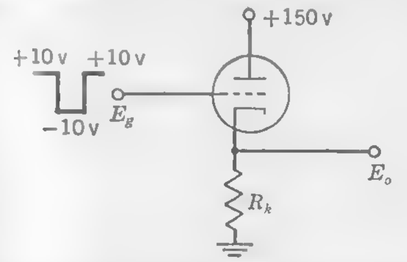


FIG. 11.43. Cathode follower with a large-amplitude input signal.

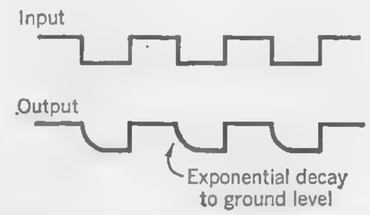


FIG. 11.44. Input and output signals for the circuit of Fig. 11.43.

feedback, stability and linearity are also excellent. It should be stressed, however, that these remarks apply only for low-level input signals at frequencies that are not too high.

If high-level signals are used as input to cathode followers some difficulties may arise; Fig. 11.43 gives an illustration of the situation. When the square wave is applied to the grid of the cathode follower, this will result in the tube being cut off during the negative half of the cycle. Then the output impedance will of course be equal to R_k. Moreover, as a result of the presence of capacitance in the load circuit, the tube will be cut off almost as soon as the input signal goes negative. If the change from the +10-volt to the −10-volt level is fast enough the cathode will not be able to follow, and the response will be an exponentially decreasing signal that tends to ground

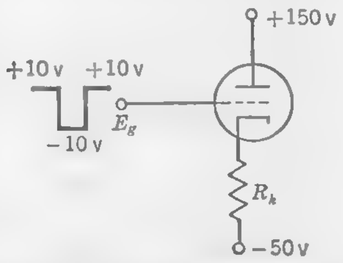


FIG. 11.45. Cathode follower with a large negative cathode return voltage.

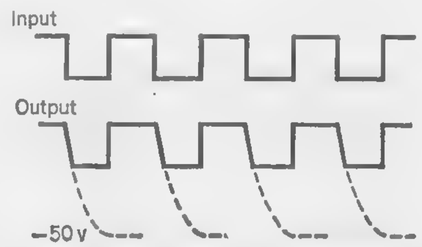


FIG. 11.46. Input and output signals for the circuit of Fig. 11.45.

potential with a time constant equal to $R_k C_k$, where C_k is the total shunt capacitance at the load. Figure 11.44 shows the input and output signals for this case.

An improvement can be obtained when the circuit of Fig. 11.45 is used. The tube is no longer cut off during the negative part of the cycle, and the output impedance is much better. Cutoff will still occur at the moment the input signal changes suddenly from +10 to −10 volts; the exponential at the output has the same time constant as before but now goes to the asymptotic value of −50 volts. The −10-volt level will, therefore, be reached relatively soon, since it is situated on a steep part of the exponential curve. Figure 11.46 gives an illustration of these facts.

Further improvement can be obtained by reducing the load capacitance, which is normally impossible, or by reducing the resistive component of the load. This can be

accomplished by using a smaller value of R_k, resulting in higher values of tube current, and perhaps by the use of a more powerful tube.

11.7.6. Cathode-coupled Amplifiers. The cathode of an amplifier tube can be used as the input electrode instead of the grid, and this may have some advantages. The fact that there is no phase reversal in the circuit is important in many cases. A disadvantage of this connection is the low value of input impedance

$$Z_i = \frac{r_p + R_p}{\mu + 1}$$

For this reason a cathode follower is very often used as an impedance-matching element, providing low input capacitance at the same time (cf. Fig. 11.47). If the tubes are identical (equal μ's and r_p's) the expression for the gain (E_1 used as the input terminal, E_2 fixed) is

$$g = \mu \Big/ \left(1 + \frac{2r_p}{R_p} + \frac{r_p(r_p + R_p)}{R_p R_k(\mu + 1)}\right)$$

Sometimes the same circuit is used with E_2 as the input terminal and E_1 fixed. The reason may be that a phase reversal is not objectionable, while the average d-c level

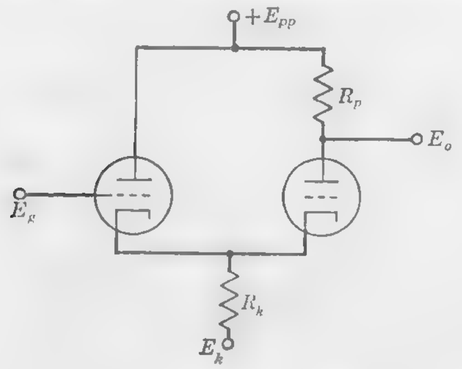


FIG. 11.47. D-C-coupled amplifier with input at the cathode applied by means of a cathode follower.

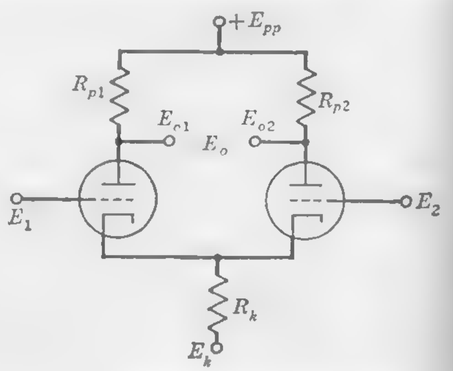


FIG. 11.48. Differential amplifier.

of E_2 is such that a higher value of cathode-supply voltage is needed than normally available. Also, the E_1 terminal may now be used to insert positive or negative feedback or to introduce a switching signal.

A somewhat different arrangement is shown in Fig. 11.48. If identical tubes and anode resistors are assumed, it can be shown that in this differential amplifier the gain from one input (E_1 for example) to the output between the two anodes (E_o) is exactly the same as it would be for a single-triode amplifier without a cathode resistor. It will be the same whether the input signal is applied at E_1 or inverted to E_2 or push-pull, one-half to each grid. The only difference is that in the last case the outputs E_{o1} and E_{o2} will be exactly push-pull, whereas in the other cases they will differ by a factor $1 + [R_p + r_p/(\mu + 1)R_k]$, which is usually very near unity. If push-pull driving is not attractive, a nearly push-pull output will still be obtained if one of the grids is held fixed, and compensation for the small deviation can be obtained by using slightly different anode resistors.

11.7.7. Regulator Amplifiers. An example of a regulator amplifier to be used in stabilized power supplies is shown in Fig. 11.49. It is a d-c-coupled amplifier with resistive feedback. Some reference voltage will be needed in this application, and this is provided by a neon glow tube T_3. It is used to determine the voltage level of the cathode of T_2. This tube amplifies the difference between its cathode potential and the grid voltage resulting from the setting of potentiometer R_2. The anode is connected to the grid of the tube T_1, which controls the flow of current through the regulator into the load. As can be seen from the diagram, the polarities are such that any

alteration in the amplitude of the output voltage E_o is opposed by the action of the regulator; this means that the regulation is operative independent of the cause of the variation of the output. These causes may be either input-voltage variations caused by variations in the a-c supply to the rectifier or load variations.

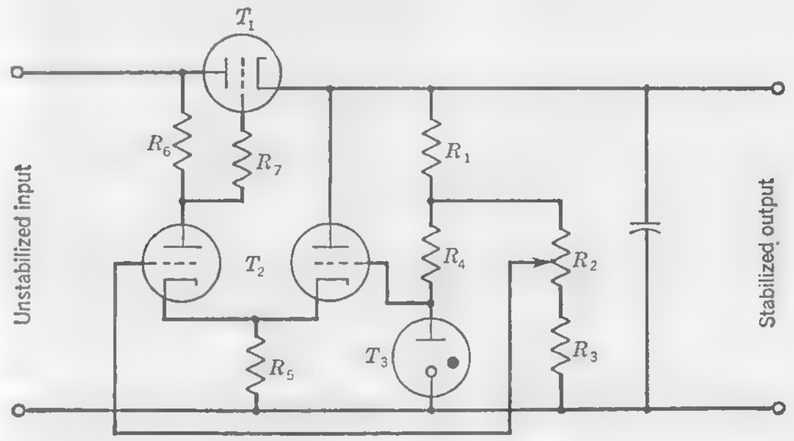


FIG. 11.49. Regulator amplifier.

The factors controlling the stability of E_o are the stability of the voltage across T_3, the constancy of the grid voltage of T_2 from the potentiometer formed by R_1, R_2, and R_3 (the use of high-stability resistors is indicated), and a high gain in the amplifier tube T_2. T_1 should be a high-current-capacity tube with high g_m and low r_p (triode connection for pentode tubes). If the total amount of load current desired is larger than can be handled by one tube, several can be connected in parallel. For optimum efficiency in this case it may be necessary to operate them with a small amount of grid current. It is then advisable to insert a cathode follower between the anode of T_2 and the grid of T_1 to give a low-impedance source for this grid current. Currents of the order of 2 amp or higher can easily be controlled by such an arrangement, and output impedances of the order of a few ohms can be obtained. The high-frequency response of the regulator is often not too good, and a large condenser across the output terminals is therefore rather useful.

11.7.8. RC-coupled Amplifiers. *Audio Amplifiers.* Figure 11.50 shows a diagram of a simple two-stage *RC*-coupled audio amplifier. The case where all bypassing (cathode and screen) is fully effective will be considered.

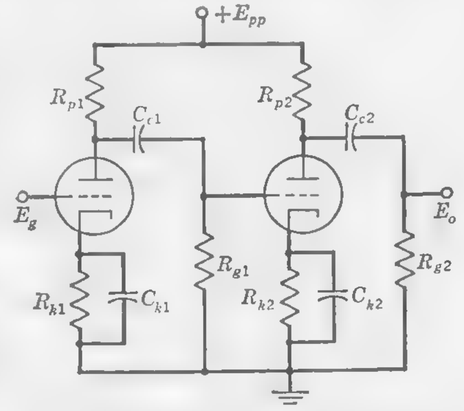


FIG. 11.50. Two-stage *RC*-coupled audio amplifier.

Figure 11.51 gives a simplified equivalent circuit (one stage) for the range of frequencies, called the midband range, where the coupling condensers can be considered as short circuits and the shunt capacitances as open circuits. As the equivalent circuit now shows no frequency-dependent elements the gain in the midband range is practically constant and is given by

$$Gm = -g_m R_h$$

where

$$R_h = \frac{1}{\dfrac{1}{r_p} + \dfrac{1}{R_p} + \dfrac{1}{R_o}} = \frac{r_p R_p R_o}{r_p R_p + r_p R_o + R_p R_o}$$

that is the equivalent of r_p, R_p, and R_o, all in parallel.

Figure 11.52 gives the equivalent circuit for frequencies above the midband range. C_s is the total shunt capacitance and consists of the output capacitance of the tube, stray wiring capacitance of the interstage coupling network, and the effective input

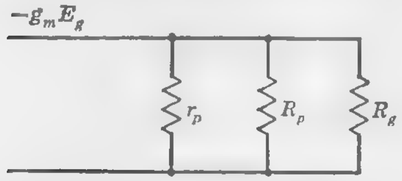


FIG. 11.51. Midband equivalent circuit for audio-amplifier stage.

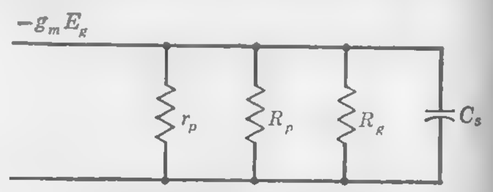


FIG. 11.52. High-frequency equivalent circuit.

capacitance of the next stage. This last quantity is not $C_{gp} + C_{gk}$ for the tube of the next stage, but as a result of the so-called Miller effect can be expressed as

$$C_t = C_{gk} + C_{gp}(1 - G)$$

where G is the gain of the next stage.

A small value of C_{gp} is, therefore, very important if a good high-frequency response is desired. C_{gp} in pentodes is much smaller than it is in triodes, and pentodes are thus invariably used in high-frequency applications.

The gain is now found to be

$$G_h = G_m \frac{1}{1 + j\omega R_h C_s}$$

Or, if amplitude and phase are considered separately,

$$|G_h| = G_m \frac{1}{(1 + \omega^2 R_h^2 C_s^2)^{1/2}} = G_m \cos \gamma$$

$$\text{phase (radians lag)} = \gamma = -\arctan \omega R_h C_s$$

This gives the phase at high frequency relative to the midband phase. (The midband

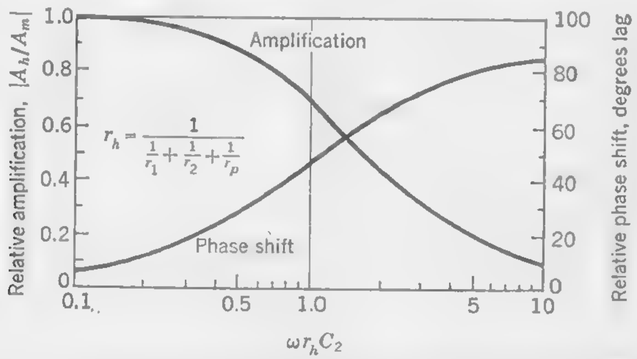


FIG. 11.53. Generic curve of relative amplification and relative phase shift of an *RC*-coupled stage at high frequencies.

itself shows a phase shift of 180 deg compared with the input.) Figure 11.53 shows a picture of the relative amplitude and phase shift at high frequencies.

The equivalent circuit at low frequency is shown in Fig. 11.54. The gain is

$$G_l = G_m \frac{1}{1 - j/\omega R_l C_c}$$

where $R_l = R_g + 1/(1/r_p + 1/R_p)$, or the series combination of R_g and the parallel combination of R_p and r_p.

$$|G_l| = G_m \frac{1}{\left(1 + \frac{1}{\omega^2 R_l^2 C_c^2}\right)^{1/2}} = G_m \cos \alpha$$

$$\text{phase} = \alpha = \arctan \frac{1}{\omega R_l C_c}$$

Figure 11.55 is the equivalent of Fig. 11.53 for low frequencies.

Compensation. The most common method to improve the high- or low-frequency response relative to the midband response is to make the load impedance dependent on the frequency. If this impedance is higher at low (or high) frequency than it is in the midband range, more gain at low (or high) frequency will be obtained; Fig. 11.56 shows some possible arrangements. In Fig. 11.56*a* the coupling impedance will be higher at low frequencies, and the low-frequency response is enhanced. In Fig. 11.56*b* a voltage-dividing action occurs for frequencies in the midband range, and high-frequency response is improved.

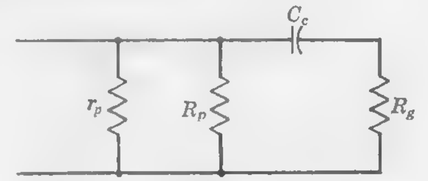


FIG. 11.54. Low-frequency equivalent circuit.

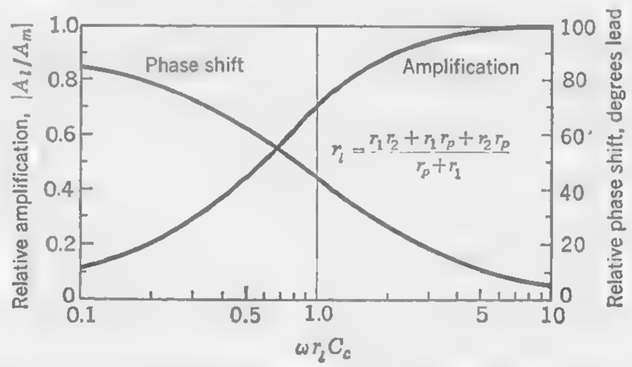


FIG. 11.55. Generic curve of relative amplification and relative phase shift of an *RC*-coupled stage at low frequencies.

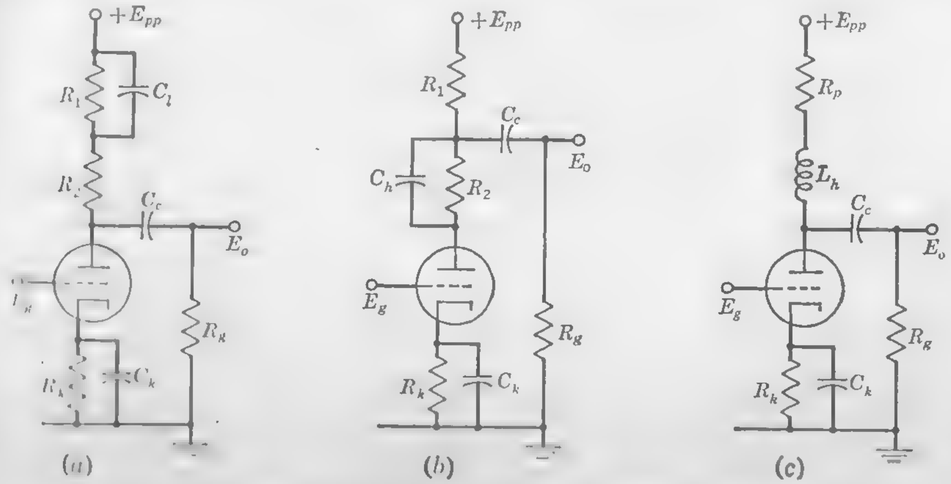


FIG. 11.56. Some circuits for low- and high-frequency compensation.

The best circuit for high-frequency compensation uses an inductance in series with R_p and is shown in Fig. 11.56*c*. If the value of this inductance is so chosen that it resonates with the effective shunt capacitance C_s at the frequency where serious loss of amplification would otherwise begin, a marked improvement can be obtained. This "peaking" inductance method will be treated in some more detail in the section on video amplifiers.

Table 11.1 gives a survey of the effect of circuit parameters on the low-, midband- and high-frequency response. A zero in the table means no influence, a + sign an improvement and a − sign a deterioration of the characteristic in each column.

Table 11.1. Effect of Circuit Parameters on Frequency Response*

Increase of	Low-frequency limit, midband	Mid-band gain	High-frequency limit, midband
g_m	0	+	0
μ	0	+	0
r_p	+	−	−
C_s	+	0	0
R_p	+	+	−
R_g	+	+	−
C_s	0	0	−

* From Reich, *Theory and Applications of Electron Tubes.*[8]

It is seen from the table that the best way of obtaining good high-frequency response (apart from compensation) is to reduce R_p and R_g. This will result in a decrease of amplification per stage, and more stages will be needed. By the use of pentodes, C_s can be reduced as well. In calculating equipment this type of amplifier is often used for such purposes as magnetic tape or drum amplifiers. Low-frequency response is not very important in most cases and is often artificially reduced to minimize trouble from hum and microphonic effects.

Video Amplifiers. This type of amplifier is usually a pulse amplifier. For this reason, the normal approach used in deriving the characteristics of audio amplifiers (amplitude and phase characteristics) is not so useful here, and the response to step functions of input voltage and rectangular pulses will be used for their analysis. The most important property of a video amplifier for many purposes is its time-delay distortion. If the input voltage is suddenly changed the output will show a response, as indicated in Fig. 11.57. The time the output needs to go from the 10 to the 90 per cent point of its final value is called the rise time. The change in the output may be followed by a certain amount of overshoot that can be expressed in volts or as a percentage of the pulse amplitude. After the first overshoot the waveform sometimes shows some more negative and positive overshoots; these damped oscillations are called overshoot oscillations. Their existence depends on the character of the interstage coupling network; they may be absent altogether.

In most applications the output pulse will be useful only when it has reached a certain amplitude. As a result of the finite rise time, this means that a useful output is obtained after a certain time delay. As a pulse may pass many stages of video amplification before being used, the total amount of time delay is a factor that must be incorporated in the design of logical switching circuits from the beginning. In many cases extra time delay is introduced into amplifiers to bring the total delay to some predetermined value, thereby simplifying the design problems arising when time delays are not exactly known.

Another property that is important in some video (and audio) amplifiers is their ability to handle signals that are very vastly different in amplitude. Such a situation may arise for instance in the case of a reading amplifier for a magnetic-drum system if the same head is used for reading and writing. The input signal to the amplifier when writing pulses are produced may have an amplitude that is one thousand times

larger than that produced at normal playback. As a result of overshoots this may cause the amplifier to be inoperative for quite a long time after the occurrence of such large signals, making the whole system useless during that time.

High-speed pentodes will be normally used in video amplifiers, since the high effective input capacitance of triodes makes them unsuitable, in most cases. The circuit

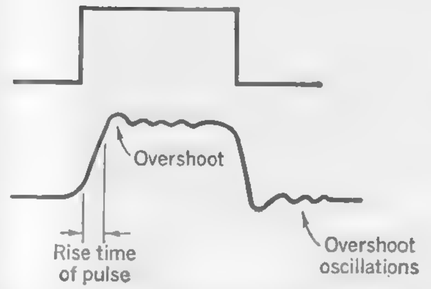


Fig. 11.57. Response of amplifier to rectangular pulses.

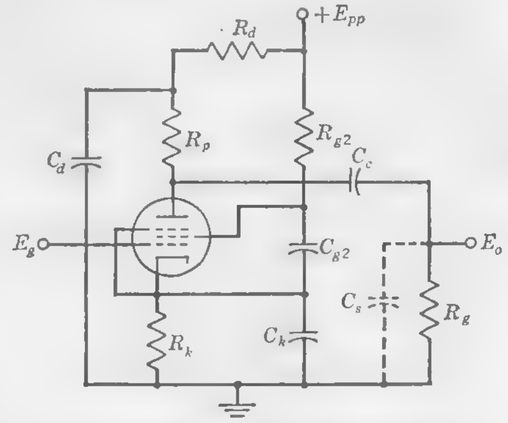


Fig. 11.58. Basic video-amplifier stage.

of Fig. 11.58 shows a basic amplifier stage with the total shunt capacitance C_s in dotted lines. Also shown is a decoupling network in the anode lead that is often incorporated to prevent coupling between stages through the power supply, and may also help to reduce hum. At very high frequencies the use of an r-f choke is advisable instead of, or in series with, R_d.

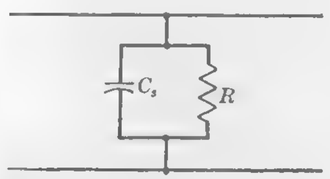


Fig. 11.59. Equivalent circuit of the amplifier of Fig. 11.58.

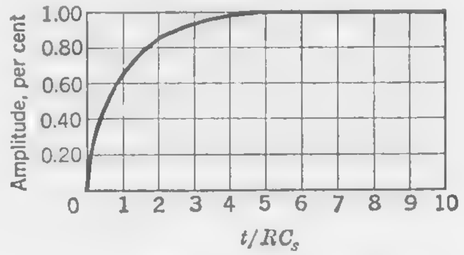


Fig. 11.60. Response to a step function of the amplifier of Fig. 11.58.

As a pentode tube is used, its plate resistance r_p may be taken as infinite. The equivalent circuit then reduces to a current generator driving the parallel combination of the load resistor R (R_p and R_g in parallel) and C_s, as indicated in Fig. 11.59.

The rise time of the output when a step function of voltage is applied at the input terminals of the amplifier is found to be

$$\text{Rise time} = 2.2\,RC_s$$

The factor 2.2 in this formula is a result of the definition of rise time given above. The values that would result from other definitions, which may be more to the point in special cases, can be found from Fig. 11.60.

As suggested in the article on audio amplifiers, the high-frequency response, and therefore the rise time, can be improved by inserting an inductance in series with R_p. The equivalent circuit for this case is shown in Fig. 11.61. This arrangement is called a shunt-peaking circuit, and it can be analyzed in terms of the parameter $m = L/R^2C$.

Response curves for various values of m (including $m = 0$, which was shown in Fig. 11.60) can be found in Fig. 11.62. It is seen that for $m = 0.5$ an improvement in rise time of a factor of 1.9 occurs in combination with an overshoot of 6.7 per cent.

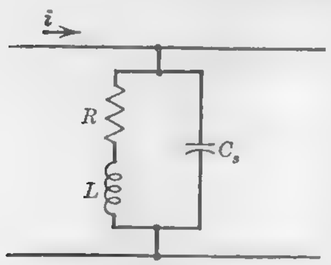


FIG. 11.61. Equivalent circuit of shunt-peaked amplifier.

More improvement can be realized by the use of more complicated, usually four-terminal, networks. They make use of the fact that C_s can be divided into an output capacitance and an input capacitance. Figure 11.63 gives an example of a circuit of this type for a ratio of 1:2. It gives an advantage of a factor 2.5 over simple RC coupling and an overshoot of only 1 per cent.

If several stages of video amplification are cascaded, the rise time of the output is worse than it is for a single stage. In an amplifier with n stages, having individual rise times $t_1, t_2, \ldots, t_n$, the final rise time is

$$t = \sqrt{t_1^2 + t_2^2 + \cdots + t_n^2}$$

In the case of identical stages, this reduces to

$$t = t_1 \sqrt{n}$$

This formula holds when no overshoot is present. If there is overshoot, the increase in rise time is somewhat less. The overshoot itself also increases approximately as the square root of the number of stages. In the case of very small overshoot (< 2 per cent), it does not increase appreciably with the number of stages.

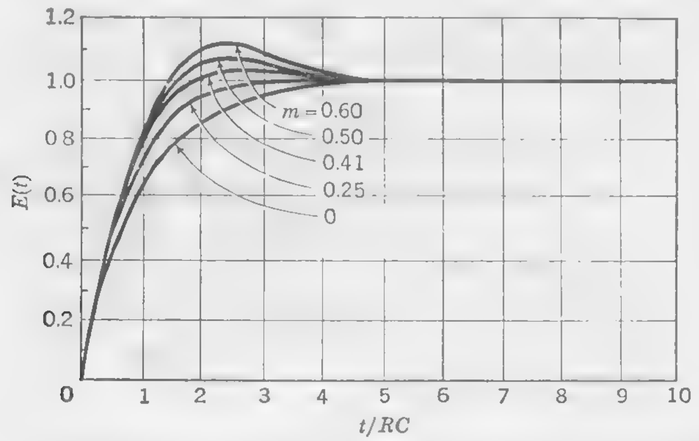


FIG. 11.62. Response of shunt-peaked circuit to a step function of current.

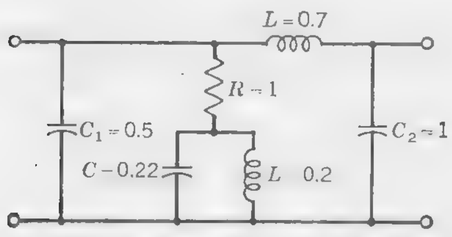


FIG. 11.63. Four-terminal compensation network.

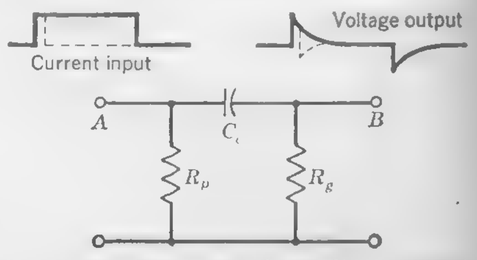


FIG. 11.64. Response of coupling network to rectangular pulse.

Because RC-coupled amplifiers do not pass the d-c component of the input signal the pulses will not preserve their flat tops as they pass through the amplifier. The

effect of the coupling circuit on the pulse top is illustrated in Fig. 11.64. R_p and R_g are the coupling and grid-leak resistor respectively. If a current pulse (a pentode tube being assumed) is applied at A the potential at A and B will rise instantaneously to the potential E, determined by the parallel combination of R_p and R_g. Then, during the flat top of the current, pulse C_c, initially uncharged, will begin to charge through the series combination of R_p and R_g, the associated time constant being equal to $(R_p + R_g)C_c$. If the pulse duration is very long compared with this time constant the condenser will charge completely, and B returns exponentially to ground potential. At the trailing edge of the pulse exactly the same thing will happen with the polarity reversed. "Differentiating" action has been obtained.

If the duration of the pulse is very short compared with the time constant, the condenser will not charge noticeably in the short time available during the flat top of the pulse and would thus act as a perfect short circuit to the pulse, introducing no distortion at all.

The most common case, in practice (unless differentiating action is desired, of course), is where the width of the pulse is fairly short compared with the time constant. Now some charging will occur, giving rise to a certain amount of "sag." At the trailing edge the pulse at B will go somewhat below ground potential and an overshoot will be the result. Similar phenomena are observed when the condenser in question is not the coupling condenser but a screen, cathode, or anode bypass condenser.

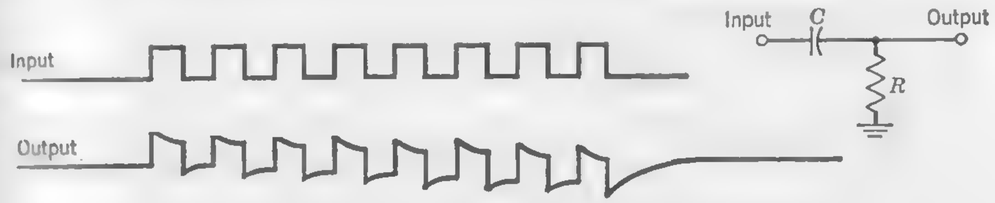


FIG. 11.65. Distortion of pulse train by RC coupling network.

A case that is often encountered in digital computer applications is that of a train of pulses separated from the next train by an interval that is long compared with both the pulse width and the RC time constant of the network. The resulting waveform is illustrated in Fig. 11.65. The condenser has completely discharged during the long interval preceding the pulse train. If a continuous train of pulses is applied at the input, each output pulse will be exactly centered around ground potential, and Fig. 11.65 shows how this situation is gradually being reestablished after the "dead" interval. In general, this behavior is very undesirable because the result probably will be that the first few pulses of the train, where the output has a higher average d-c potential, will be better amplified by the succeeding stage than those that come later, introducing extra distortion.

To prevent this happening some means of d-c restoration must be employed. This can be done by making the RC time constant during the top flat portion of the pulse high, and during the negative overshoot small. The best way to accomplish this is to use some kind of limiter, usually a diode or germanium crystal. The circuit, together with the associated waveforms, is shown in Fig. 11.66. Some overshoot will still be present, since the forward resistance of the diode is not zero, but the time constant will be so small that the equilibrium situation will be restored long before the arrival of the next pulse; therefore, all pulses are now treated in the same way, independent of the length of the preceding interval. A disadvantage of this circuit is that it may pass pulses of one polarity (positive in the circuit of Fig. 11.66) only. The name given to circuits of this type is "clamping" circuits.

A somewhat similar circuit, called a "clipping" circuit, can be used to provide pulses with a better flat top at the cost of amplitude. Figure 11.67 shows this arrangement, together with input and output waveforms. To obtain good clipping, the resistor R_s must be large compared with the forward resistance of the diode.

As a result of the unsharpness of the knee in the voltage-current characteristics of diodes in the vicinity of zero voltage, both clamping and clipping circuits will be

effective only for signal amplitudes of the order of some volts or more. At very small amplitudes, diodes behave more or less as linear resistors.

Overshoots can become very troublesome when tubes are driven into the grid-current region by positive pulses of appreciable amplitude. When this happens, the grid-leak resistor becomes effectively zero, and as a result of the clipping action of the

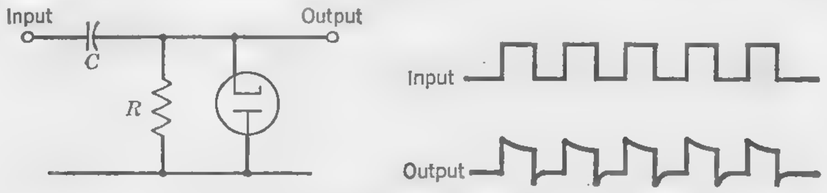


FIG. 11.66. D-C restoration and associated waveforms.

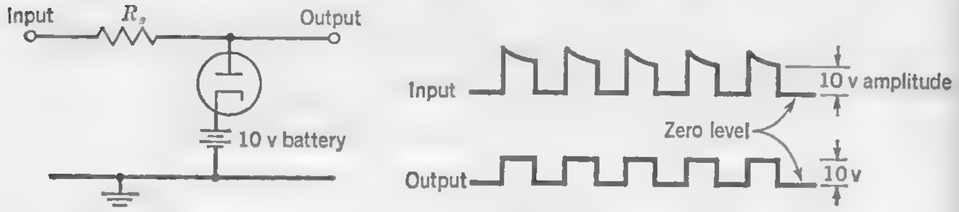


FIG. 11.67. Clipping with a diode and input and output waveforms.

tube, a nice flat pulse is obtained at the output. The coupling condenser is charged by grid current with the very short time constant provided by the low value of grid-cathode resistance. At the trailing edge of the pulse a severe negative overshoot occurs that decays with the normal time constant of the coupling circuit, as the tube is now probably cut off. This means that the amplifier will be insensitive to small signals following the large pulse as long as the cutoff condition lasts.

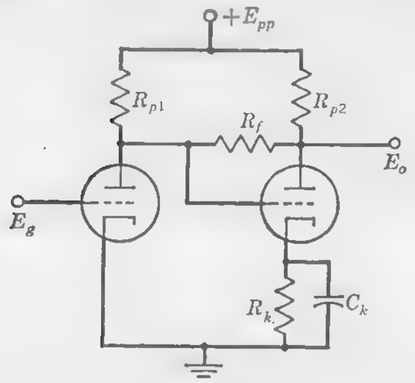


FIG. 11.68. Circuit to reduce dead time of amplifier by elimination of coupling condensers and negative feedback.

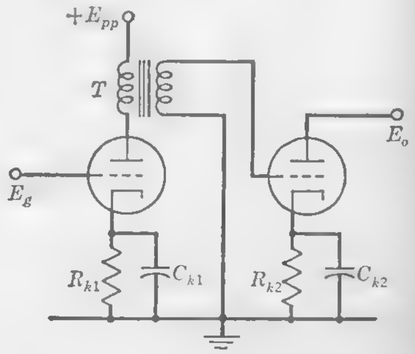


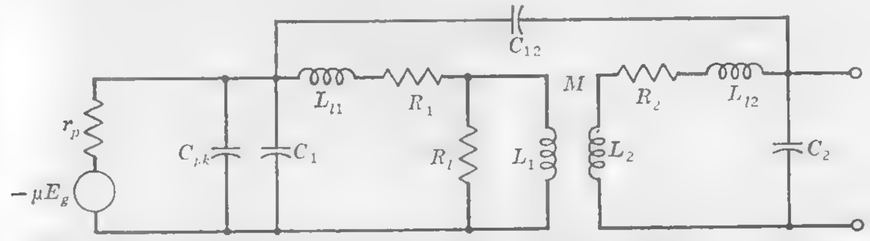
FIG. 11.69. A transformer-coupled audio-amplifier stage.

To obtain fast recovery one can either try to prevent overshoots altogether or to make their time constants so small that they do not matter. The first possibility means that very long time constants must be used throughout the amplifier, the second that very short time constants are indicated everywhere. It should not be concluded that the best procedure is to choose some intermediate value for the time constants, because this will give the worst possible results.

Overshoots produced by long time constants can usually be sufficiently attenuated by having a short time constant somewhere farther on; the most important thing is to prevent any stage from being overdriven, since this will result in a change of ampli-

fication that cannot be compensated by succeeding time constants. The use of direct coupling between stages is sometimes possible; the elimination of the condenser makes overshoots impossible. Inverse feedback also helps in preventing overshoot. Figure 11.68 shows a possible circuit arrangement.

11.7.9. Transformer-coupled Amplifiers. Figure 11.69 shows a diagram of a transformer-coupled audio amplifier. A stepup or stepdown transformer (turns ratio



r_p = Plate resistance
C_{pk} = Output capacitance
C_1 = Primary distributed capacitance
R_1 = Primary resistance
R_l = Loss resistance of the core
L_1 = Primary inductance
L_2 = Secondary inductance
M = Mutual inductance between windings
R_2 = Secondary resistance
C_2 = Secondary distributed capacitance
C_{12} = Distributed capacitance between primary and secondary
l_1 = Primary stray inductance
l_2 = Secondary stray inductance

FIG. 11.70. Exact equivalent circuit of transformer-coupled audio-amplifier stage.

n) is used as the coupling element. The exact equivalent circuit is quite complicated, as shown in Fig. 11.70. For many purposes the simplified version shown in Fig. 11.71 may be used. The gain is found to be

$$G = \frac{L_1 n \mu k}{C_{eq}(R_1' + j\omega L_1)\left[R_2 + j\left(\omega L_2 - \frac{1}{\omega C_{eq}}\right) + \frac{k^2 L_1 L_2 \omega^2}{R_1^2 + j\omega L_1}\right]} \tag{11.10}$$

where k is the coefficient of coupling: $k = M/\sqrt{L_1 L_2}$. At low frequencies $1/\omega C_{eq}$ is very large and, since k is approximately unity in a well-designed transformer, Eq. (11.10) reduces to

$$G_l = \frac{j\omega L_1}{R_1' + j\omega L_1}\, n\mu$$

When ω is increased, this can be further reduced to an expression that does not contain frequency-dependent elements. The midband gain is

$$G_m = n\mu$$

G_l may now be rewritten as

$$G_l = G_m \frac{j\omega L_1}{R_1' + j\omega L_1}$$

$$|G_l| = G_m \frac{\omega L_1}{[R_1'^2 + (\omega L_1)^2]^{1/2}} = G_m \cos\gamma$$

$$\text{phase} = \gamma = \arctan \frac{R_1'}{\omega L_1}$$

is the phase shift at low frequency relative to the midband phase. The midband output may be either in phase with the input or 180 deg out of phase.

At high frequencies ($R_1 \ll \omega L_1$) the leakage inductances must be taken into account. They can be replaced by an equivalent leakage inductance referred to the secondary: $L_{eq} = n^2 L_{l1} + L_{l2}$. When the resistance on the primary side is also referred to the secondary, it becomes $R_{eq} = R_2 + n^2 R_1'$.

Then we find

$$G_h = \frac{n\mu}{j\omega C_{eq}\left[R_{eq} + j\left(\omega L_{eq} - \frac{1}{\omega C_{eq}}\right)\right]} \tag{11.11}$$

This expression has a maximum value when $\omega L_{eq} = 1/\omega C_{eq}$, that is, at the frequency $\omega_m = 1/\sqrt{L_{eq}C_{eq}}$, where the equivalent leakage inductance resonates with the equivalent distributed capacitance. Introducing this quantity into Eq. (11.11),

$$G_h = \frac{G_m}{j\frac{\omega}{\omega m}\frac{1}{Q} - \left(\frac{\omega}{\omega m}\right)^2 + 1}$$

where $Q = 1/R_{eq}\sqrt{L_{eq}/C_{eq}}$.

At the frequency $\omega = \omega_m$, $G_h = -jG_mQ$. The amplitude at $\omega = \omega_m$ is Q times the midband value and the phase is lagging by 90 deg. A value of Q in the vicinity of 1 is usually best in practice.

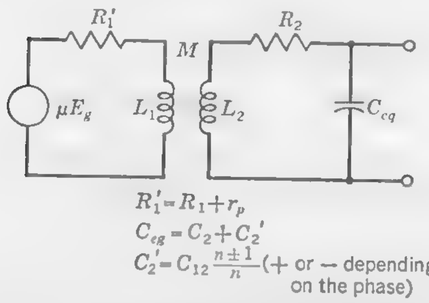


FIG. 11.71. Simplified equivalent circuit for transformer-coupled audio-amplifier stage at midband and low frequencies.

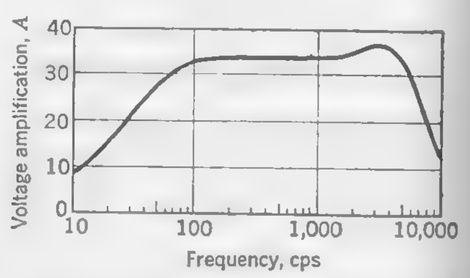


FIG. 11.72. Frequency-response curve of a transformer-coupled audio-frequency amplifier.

Figure 11.72 shows a typical amplitude-response curve. If a step-up transformer is used as a coupling element the step-up ratio is normally not more than 1:3, because a very large turns ratio combined with a high primary inductance (necessary for good low-frequency response) will lead to very many turns, high distributed capacitance and poor high-frequency response. The total voltage amplification that can be obtained with triode transformer-coupled stages is, therefore, limited to about 50 per stage compared with a value of 100 for pentode resistance-capacitance-coupled stages. Hence, they are only used for special applications—obtaining push-pull output by using a center tap on the secondary, or matching of very low impedance loads.

11.7.10. Tuned Amplifiers. In tuned amplifiers the load impedance has the form of a resonant circuit. Tuned amplifiers can be divided in narrow band amplifiers, used when the required bandwidth is a small fraction of the carrier frequency, and bandpass (usually double tuned) amplifiers where the bandwidth is of the same order of magnitude as the carrier frequency. The diagram of a single tuned amplifier is shown in Fig. 11.73. (D-C connections have been omitted.) The tubes used in this type of amplifier are of the same type as those used in video amplifiers, that is, high-speed pentodes.

The resonant frequency of the load circuit $f_0 = 1/2\pi\sqrt{LC}$. The gain is given by

$$G = g_m Z$$

where Z is the impedance of the load circuit. Since g_m is a constant, the amplitude characteristic of the amplifier is similar to the resonance curve of the load circuit (Fig. 11.74).

The gain at frequency f_0 is $G_r = g_m R$, where R is the total circuit resistance, containing the combined effects of the plate resistance, the load resistor, and the shunt losses of L and C. The bandwidth of the circuit (3 db or 70.7 per cent points) is $B = 1/2\pi RC$. It is seen from the formulas that a large value of bandwidth goes together with a small

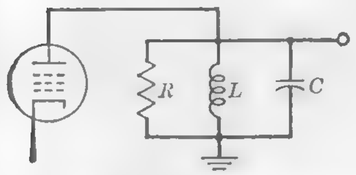


FIG. 11.73. Single-tuned amplifier stage with all d-c connections omitted.

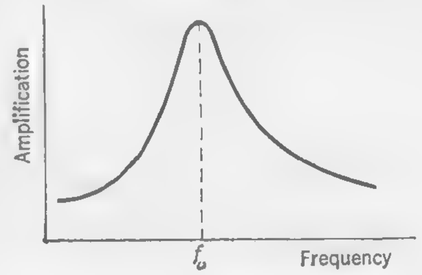


FIG. 11.74. Amplitude-response curve of single-tuned amplifier.

value of gain; bandwidth may be improved at the cost of gain. The gain-bandwidth product can therefore be taken as a figure of merit: $\alpha = g_m/2\pi C$. The bandwidth of an amplifier consisting of a number of single-tuned stages, all tuned to the same resonant frequency, can be shown to be approximately $1/1.2\sqrt{n}$, where n is the number of stages. This factor can be improved, in fact reduced to 1, by tuning some of the stages to a slightly different frequency (staggering), but a discussion of these methods is outside the scope of this book.

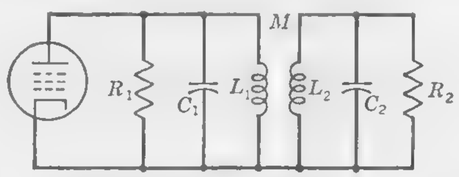


FIG. 11.75. Double-tuned amplifier stage with d-c connections omitted.

Figure 11.75 is the equivalent of Fig. 11.73 for a double-tuned circuit. The gain can be written as

$$G = \frac{g_m\sqrt{R_1R_2}\,s}{\sqrt{(1+s^2)^2 - 2\left(s^2 - \frac{b}{2}\right)v^2 + v^4}}$$

where $s = k\sqrt{Q_1Q_2}$

$K = \dfrac{M}{\sqrt{L_1L_2}}$ the coupling coefficient

$b = \dfrac{Q_1}{Q_2} + \dfrac{Q_2}{Q_1}$

$v = \sqrt{Q_1Q_2}\left(\dfrac{\omega}{\omega_0} - \dfrac{\omega_0}{\omega}\right)$ the modified frequency variable

Q_1 and Q_2 are the Q's of the primary and secondary tuned circuits $\omega = 2\pi f$, and $\omega_0 = 2\pi f_0$.

The expression for midband gain is

$$G_m = \frac{g_m\sqrt{R_1R_2}\,s}{1 + s^2}$$

Maximum midband gain is obtained when $s = 1$;

$$G_{max} = \tfrac{1}{2} g_m \sqrt{R_1 R_2}$$

In this case $k = 1/\sqrt{Q_1 Q_2}$, and this is called critical coupling. If the conditions for maximal flatness of the response curve are solved we find $s^2 = \frac{1}{2}b$ and

$$k = \frac{1}{\sqrt{Q_1 Q_2}} \sqrt{\frac{b}{2}}$$

Now the so-called transitional coupling is obtained. This name is derived from the fact that for this value of K the response curve shows only one peak, whereas for larger values of K two peaks are present. If the primary and secondary circuits have equal Q's, $b = 2$ and transitional and critical coupling are the same thing.

The bandwidth with transitional coupling is

$$B = \frac{\sqrt{1 + \sqrt{1 + s^2}}}{2\pi \sqrt{C_1 C_2} \sqrt{R_1 R_2}}$$

For n transitionally coupled double-tuned stages the bandwidth is $\sqrt[4]{2^{1/n} - 1}$ times that of a single stage. It should be noted that the formulas given here apply strictly to the case of high Q's. The low-Q case is much more difficult, but the high-Q formulas can be used as useful approximations and furnish a starting point for experimental methods.

Tuned amplifiers have limited applications in computer equipment. They can be used to amplify acoustic delay-line signals using 10–20-mc carrier frequencies. Large bandwidth is necessary in high pulse-frequency applications, and relatively low Q values will have to be used. An important characteristic in this case is the rise time of the output on the application of a stepfunction of carrier-frequency signal at the input. As a general rule it can be said that quite independent of the type of circuit used in the interstage-coupling network (single-tuned or transitionally coupled double-tuned), the rise time (10 to 90 per cent points) can be taken as 0.7 times the reciprocal of the 3-db bandwidth.

11.7.11. Transformer-coupled Drivers. Pulse amplifiers using transformers in the anode circuit are extensively used in computer circuits. Some of the advantages of this circuit are

1. Reshaping, power amplification, and impedance matching can be realized by the use of only one tube and transformer.
2. Pulses of both polarities can be obtained from the transformer output.
3. By using a pentode tube with the load line in a very low position (bottoming), very stable operation can be obtained independent of the tube characteristics.
4. If necessary the backswing of the transformer, when the current pulse through the tube has died out, can be used as a convenient means of obtaining a pulse delay at the same time.
5. No anode current (or a very small amount) flows when no pulses are applied to the input.

Among the desirable properties for tubes used in these amplifiers are high transconductance, low input and output capacitances, high current-carrying capacity and high dissipation. Small power pentodes are therefore normally used. The transformers employed usually have ferrite cores, and correct operation at frequencies in excess of 1 mc can be obtained.

Figure 11.76 shows a basic circuit diagram. If high secondary current is wanted, a high turns ratio must be used. On the other hand, this means that a high primary voltage will be needed, and then the plate capacitance must be charged to a correspondingly high potential. When the desired secondary amplitude and rise time are known, the best value for the turns ratio can be computed. The analysis shows it should be so chosen that during the rise time of the pulse half the average current is used to charge the anode capacitance.

The grid of the tube should be driven positive if dissipation ratings allow this, because the secondary current during the rise will be proportional to the square of the plate current; therefore any increase in plate current is very useful.

To obtain a flat-topped output pulse the change of B in the transformer per unit of time must be constant. This means that during the pulse more and more magnetizing current will be needed, and the tube must be able to supply this while the operating point must not move beyond the knee of the $e_p - i_p$ characteristics (cf. Fig. 11.76).

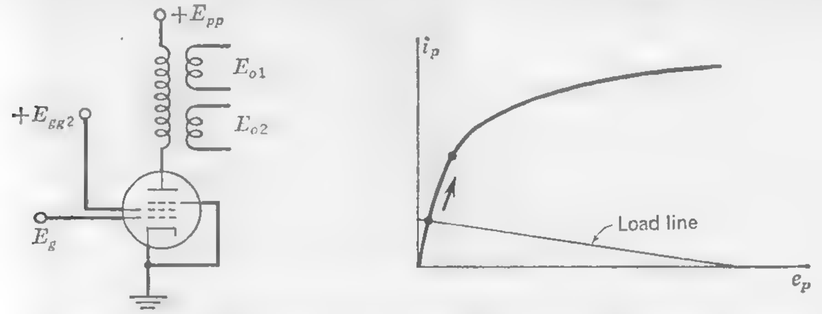


Fig. 11.76. Basic circuit of transformer-coupled driver and load-line position on e_p-i_p characteristics.

The damping introduced by the load at the secondary must be sufficient to prevent the occurrence of more than one overshoot at the termination of the tube pulse. Any further overshoots would interfere with the operation of the circuit during the next pulse time.

Although the switching mechanism of rectangular hysteresis cores is quite different from the operation of normal transformer materials, similar reasonings apply to the pulse amplifiers used to drive them. Here too, constant voltage output from the tubes by the use of the bottoming characteristics of a pentode is often useful to obtain constant currents in the secondary windings, which are essential for the operation of half-current, coincident-type memories. Higher values of primary current (200 ma) are often desired, and powerful output pentodes have to be used.

11.8. PARAMETRIC OSCILLATORS

Parametric subharmonic phase-locked oscillators can be used to store binary information in terms of its two possible phases of oscillation. The physical appearance of

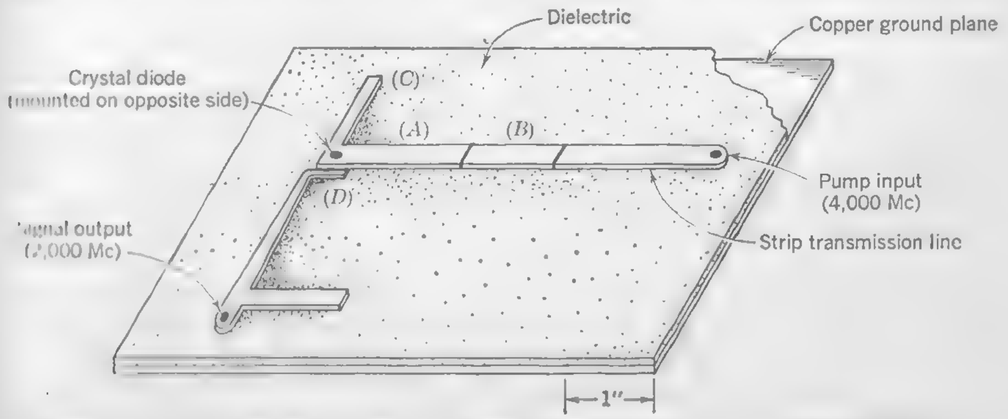


Fig. 11.77. Microwave subharmonic phase-locked oscillator. (*Courtesy of RCA.*)

such an oscillator operating at microwave frequencies is shown in Fig. 11.77. The tank circuit, pump frequency, and output signals are shown in the various parts of Fig. 11.78.

If the pump frequency is interrupted, then relatively little energy is needed to establish one particular phase when the pump frequency again starts.

Parametric oscillators using a nonlinear inductance have been used in computers in Japan.[1] A junction *p-n* diode whose capacity is a function of back voltage can be used making possible circuits operating at kilomegacycles instead of megacycles.

The reason for the build-up of oscillations at half the pump frequency can be explained as follows: At the time of maximum charge on the condenser the capacity is reduced by the increase in potential due to the pump voltage. The capacity is restored to its

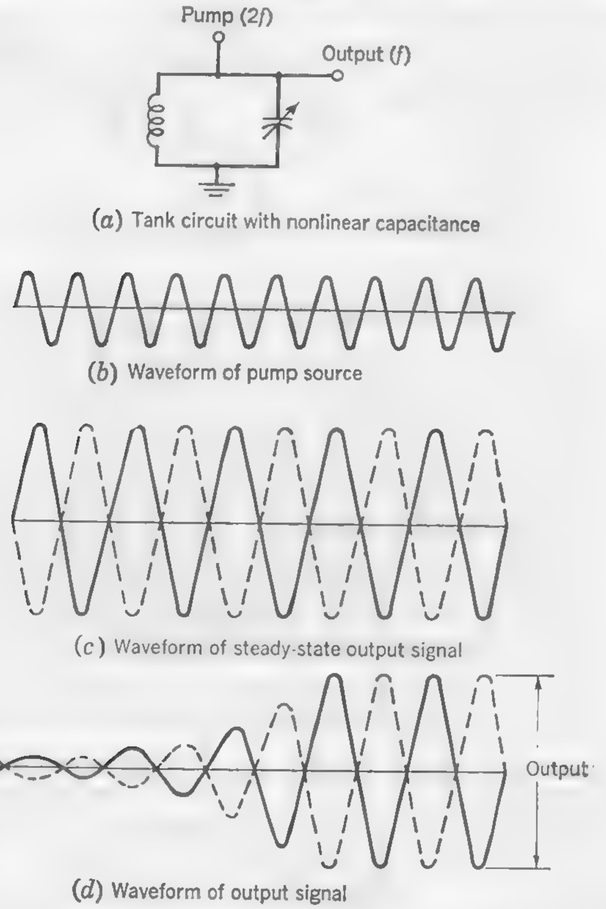


FIG. 11.78. Principle of parametric phase-locked oscillator. (*Courtesy of RCA.*)

original value when its charge is zero. The work required to reduce the capacity increases the energy in the condenser, and this happens for each half cycle of the oscillation. If this addition exceeds or equals the losses of the circuit the oscillation will be maintained. Upper limits on the amplitude of the oscillation may be caused by detuning or by conductance in the diode.

The addition of energy occurs once each half cycle and thus can maintain an oscillation in either of two phases which are "locked" to the phases of the pump frequency. These two phases can denote a binary 0 or 1.

References

1. Goto, E.: On the Application of Parametrically Excited Nonlinear Resonators, *J. Institute Electrical Communication Engineers of Japan*, **38**:770–775 (October, 1955).
2. Guterman, S. S., and W. M. Carey, Jr.: A Transistor-magnetic Core Circuit: A New Device Applied to Digital Computing Techniques, *IRE Convention Record*, part 4, New York: 84–94, 1955.

3. Kaufmann, Henry W.: Semi-conductor Diode Amplifier Considerations, *IRE Convention Record*, part 4, New York: 146–149, 1955.
4. Millman, Jacob, and Herbert Taub: *Pulse and Digital Circuits*, McGraw-Hill Book Company, Inc., New York, 1956.
5. Nienburg, Raymond E.: Reliability of an Air Defense Computing System: Circuit Design, *IRE Trans. on Electronic Computers*, **EC-5** (4): 227–233 (December, 1956).
6. Onyshkevych, L. S., W. F. Kosonocky, and A. W. Lo: Parametric Phase-Locked Oscillator Characteristics and Applications to Digital Systems, *IRE Trans. on Electronic Computers*, **EC-8** (3): 277–286 (September, 1959).
7. Raymond, G. A.: A Transistor-circuit Chassis for High Reliability in Missile-guidance Systems, *Proc. Eastern Joint Computer Conf.*, Washington, D.C.: 132–135. Dec. 9–13, 1957,
8. Reich, H. J.: *Theory and Applications of Electron Tubes*, McGraw-Hill Book Company, Inc., New York, 1944.
9. Rosenberger, G. B.: A Cryogenic Oscillator, *IBM J. Research Development*, **3**(2): 189–190 (April, 1959).
10. Valley, G. E., and H. Wallman (eds.): *Vacuum Tube Amplifiers*, MIT Radiation Laboratory Series, vol. 18, McGraw-Hill Book Company, Inc., New York, 1948.

Section 12

MEMORY DEVICES

By

ISAAC L. AUERBACH, *President and Technical Director, Auerbach Electronics Corporation, Philadelphia, Pa.*

ALBERT S. HOAGLAND, *Senior Engineer, International Business Machines Research Laboratory, San Jose, Calif.*

ARTHUR W. HOLT, *Chief Electronic Engineer, Rabinow Engineering Company, Washington, D.C.*

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

CHARLES F. PULVÁRI, *Professor, Department of Electrical Engineering, The Catholic University of America, Washington, D.C.*

RAYMOND STUART-WILLIAMS, *Technical Director, Ampex Computer Products Company, Ampex Corporation, Culver City, Calif.*

FREDERIC C. WILLIAMS, *Professor, Department of Electrical Engineering, The University, Manchester, England.*

CONTENTS

12.1. Introduction 12–2
12.2. Flip-flops 12–3
12.2.1. The Vacuum-tube Flip-flop 12–3
12.3. Ferromagnetic Shift Registers 12–5
12.3.1. Basic Operation 12–5
12.3.2. Transfer Loops 12–7
12.3.3. Noise Generation 12–11
12.3.4. Typical Magnetic Shifting Registers 12–13
12.3.5. Magnetic-core Characteristics 12–14
12.3.6. Diode Characteristics 12–15
12.3.7. Drivers for Magnetic-core Circuits 12–16
12.4. Mercury and Quartz Delay Lines 12–16
12.5. Magnetic Memory Systems 12–17
12.5.1. The Magnetic-recording Process 12–19
12.5.2. The Magnetic-recording Theory 12–20
12.5.3. Magnetic-head Design 12–22
12.5.4. Recording Techniques 12–28
12.5.5. Head Selection and Switching 12–31
12.5.6. System Descriptions 12–31
12.6. Williams-tube Memory System 12–34
12.6.1. Basic Arrangement 12–34
12.6.2. Technological Theory 12–34
12.6.3. Practical Systems 12–37
12.6.4. Operating Factors 12–38
12.6.5. Conclusion 12–41
12.7. Magnetic-core Storage and Switching Techniques 12–41
STORAGE CORES
12.7.1. Characteristics of Storage Materials 12–42

12.7.2. Switching Times of Storage Materials..... 12–42
12.7.3. Types of Storage Materials 12–43
12.7.4. Core Geometry..... 12–45
12.7.5. Typical Ferrite Cores.... 12–47
12.7.6. Testing Storage and Switch Cores..... 12–51

STORAGE TECHNIQUES

12.7.7. Principles of Coincident-current Selection..... 12–52
12.7.8. Operation of a Number of Matrices in Parallel..... 12–55
12.7.9. Behavior of Minor Loops in Storage Cores..... 12–56
12.7.10. Linear or Word Selection 12–58
12.7.11. Matrices and Sense Windings..... 12–60

SWITCHES AND SWITCHING TECHNIQUES

12.7.12. Principles of Magnetic-switch Operation..... 12–64
12.7.13. Magnetic-core Switches.. 12–65
12.7.14. Diode and Transistor Switches..... 12–79

STORAGE UNITS AND SYSTEM DESIGN

12.7.15. Coincident-current Storage Units..... 12–80
12.7.16. Methods of Driving Coincident-current Storage Units..... 12–81
12.7.17. Circuits Peculiar to Core-storage Units..... 12–86
12.7.18. Typical Magnetic-core Storage Units..... 12–90
12.7.19. System Design of Coincident-current Core-storage Units..... 12–98
12.7.20. Design of Core-storage Units for Adverse Environments..... 12–100
12.7.21. Testing of Magnetic-core Storage Units..... 12–101
12.7.22. Maintenance of Coincident-current Storage Units..... 12–102
12.7.23. Nondestructive Read-out 12–103
12.7.24. Future Core-storage Units 12–105
12.8. Ferroelectrics and Their Memory Properties..... 12–106
12.8.1. $BaTiO_3$ Single Crystals.. 12–107
12.8.2. The Bistable Storage Cell 12–107
12.8.3. The Bistable Condenser as a Circuit Element.... 12–110
12.8.4. Transpolarizer..... 12–112
12.8.5. Ferroelectric Switches... 12–114
12.8.6. A Shift Register..... 12–115
12.9. Diode-capacitor Memory System..... 12–116
12.9.1. Introduction..... 12–116
12.9.2. Basic Cell..... 12–117
12.9.3. Arrangement of Cells in Matrix..... 12–118
12.9.4. Access Circuitry—The Transformer-diode Coincidence Gate..... 12–119
12.9.5. Typical Silicon Diode Specification..... 12–120
12.9.6. Holding Equations..... 12–120
12.9.7. Writing Equations..... 12–122
12.9.8. Reading Equations..... 12–123
12.9.9. Block Diagram of the System..... 12–124
12.9.10. Access Drivers..... 12–124
12.9.11. Gating Amplifiers..... 12–125
12.9.12. Checking and Maintenance..... 12–125
12.9.13. Physical Construction... 12–125

12.1. INTRODUCTION

The first high-speed electronic computer to be put into operation was the ENIAC (electronic numerical integrator and computer). It made use of flip-flop circuits to store information. A flip-flop is a bistable electronic circuit capable of storing one binary digit (bit) of information. The early circuits used vacuum tubes; more recently transistors and magnetic cores have been used (cf. Sec. 12.3). Before the ENIAC was put into operation the builders realized that a large low-cost-per-bit memory was required in digital computers. They hit upon the use of acoustic delay lines to store long trains of pulses. The delay lines had been developed for range-measurement purposes in radar systems. Another radar device, the moving-target indicator, led to the Williams-tube memory. The perfection of magnetic materials with square hysteresis loops led to the development of magnetic-core memories. A similar square-hysteresis-loop phenomenon was discovered in ferroelectric materials, and considerable effort has been spent in trying to perfect condenser-type memories. Each of these techniques of storing information will be discussed in the sections of this chapter.

12.2. FLIP-FLOPS

A flip-flop is any circuit which has two stable states. If the two states are referred to as the "set" and "reset" states, then it is conventional to say that when in the set state the flip-flop stores a "1" and when in the reset state it stores a "0." There may be three kinds of input to a flip-flop: (1) a set input which is such that if the proper signal appears on it the flip-flop remains in, or is changed to, the set state; (2) a reset input capable of placing the flip-flop in the reset state; and (3) a change input by means of which it is possible to change the state of the flip-flop from reset to set or from set to reset. A flip-flop with a change input is sometimes referred to as a binary unit counter.

12.2.1. The Vacuum-tube Flip-flop. Figure 12.1 shows a circuit of a vacuum-tube flip-flop. Note that it consists of two d-c-coupled amplifiers (cf. Fig. 11.38) with the output of the second connected to the input of the first. The two stable states consist of one tube fully conducting and the other cut off, or vice versa. During the transition state it is a two-stage amplifier with positive feedback.

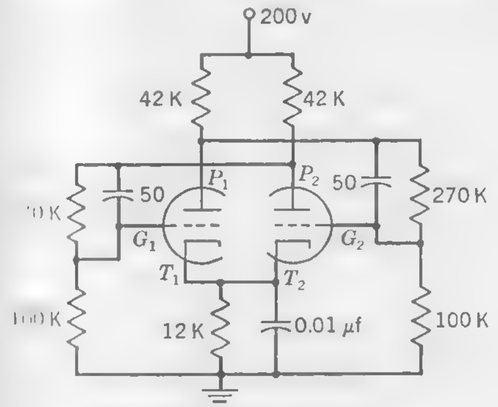


Fig. 12.1. 100-kc flip-flop.

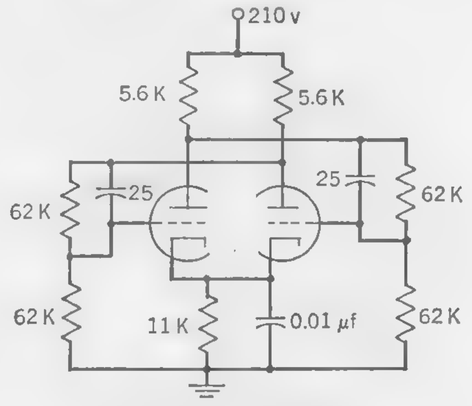


Fig. 12.2. 1-Mc flip-flop.

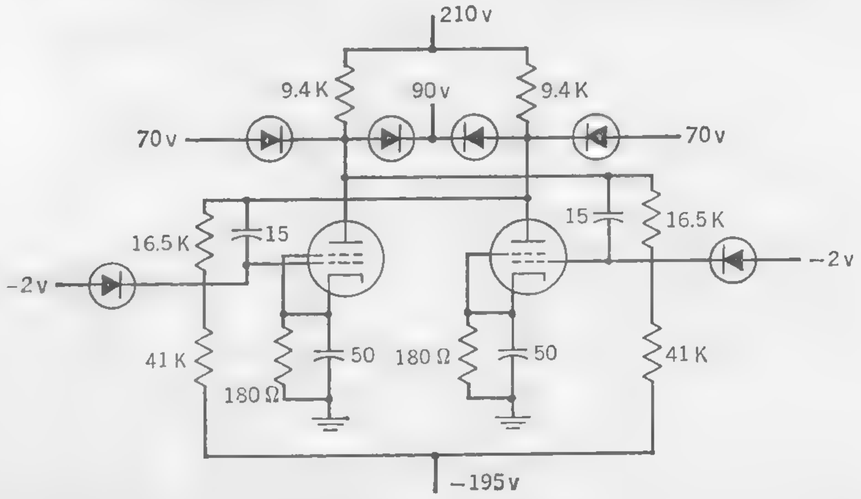


Fig. 12.3. 10-Mc flip-flop.

Another way to consider the circuit is as follows: Assume that tube $T1$ is conducting. Then $P1$ is at a low potential forcing $G2$ to be low (below cutoff). Thus tube $T2$ is not conducting and plate $P2$ is at a high potential. This in turn holds $G1$ high, causing tube $T1$ to conduct. A negative pulse on $G1$ or a positive pulse on $G2$ will cause the circuit to change its state to where $T2$ is conducting and $T1$ is cut off.

Without the plate-to-grid coupling condensers the circuit still has two stable states. With the coupling condensers the high-frequency response of the feedback loop is much better and the circuit changes its state much more quickly. That is, shorter pulses are sufficient to set or reset the circuit.

Consider now what happens if a short positive pulse is applied to the cathode or if a negative pulse is applied to both grids simultaneously. The effect is to cut off both tubes. Start with the assumption that $T1$ is conducting; then the potential across the condenser from plate $P1$ to $G2$ is less than that from $P2$ to $G1$. When the cathode, or grid pulses, are applied, anode $P1$ rises toward the supply voltage (that is, $P1$ approaches the same level as $P2$). Thus, when the input pulse ceases, grid $G2$ is higher than $G1$, and $T2$ begins to conduct before $T1$. Note that for reliable operations the input pulse must be of the correct shape. If it is too long the coupling condensers change their charge so that $G1$ and $G2$ are the same potential and it becomes indeterminate which tube will conduct when the input pulse ceases. If the input pulse is of insufficient size or duration to cut off $T1$, no triggering action occurs.

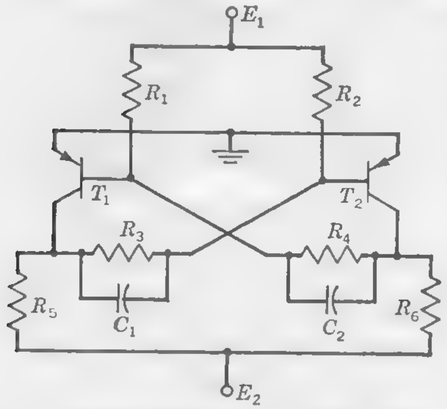


FIG. 12.4. Eccles-Jordan transistor flip-flop.

The stability of the flip-flop depends upon how insensitive the circuit is to small grid signals. Stability can be increased by making the grid of the nonconducting diode go farther below cutoff. Stability can also be attained by holding the grid against a clamp even though the tube is still conducting.

Transition speed depends upon the size of the plate resistors and the current drawn in the tubes. The discussion for d-c-coupled amplifiers (cf. Sec. 11.7) applies here.

Resolution is measured as to how close two pulses may be and still have the circuit respond to each pulse. Resolution is improved by reducing the coupling condensers and/or reducing the size of the resistors in the coupling networks.

The two factors which require most attention from the circuit designer are stability and resolution time. Usually improvement of resolution time decreases stability.

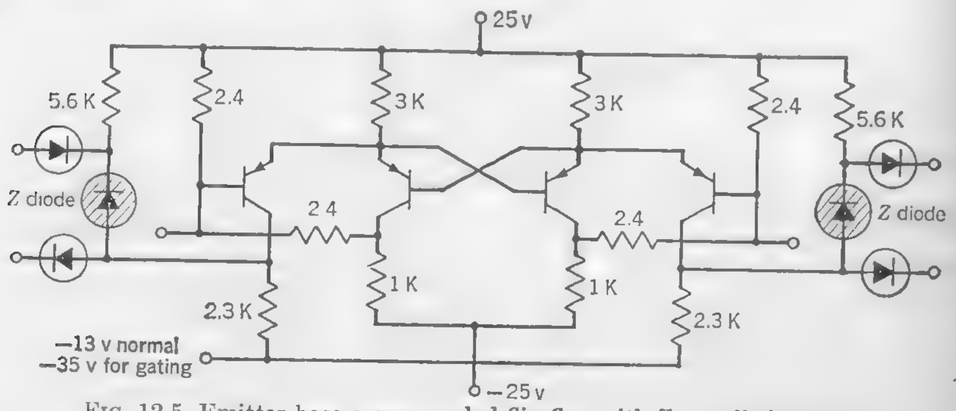


FIG. 12.5. Emitter-base cross-coupled flip-flop with Zener diode gates.

Figures 12.1, 12.2, and 12.3 show three flip-flop circuits designed to operate at 100 kc, 1 Mc, and 10 Mc, respectively. The speed in the circuit of Fig. 12.3 has been obtained by reducing the coupling capacitors (using pentodes with no miller capacity to make this possible) and by charging and discharging the coupling capacitors by the clamping diodes.

Examples of transistor flip-flops are shown in Figs. 12.4 and 12.5.

12.3. FERROMAGNETIC SHIFT REGISTERS*

12.3.1. Basic Operation. *Introduction.* A shifting register is, in general, a storage device which records the individual digits of a number in a serial arrangement of memory elements and provides additionally a mechanism for transfer to adjacent positions in the register. Adaptable to both serial or parallel insertion and delivery of data, this versatile logical element finds extensive use in computer organization.

Of the various bistable components used in the design of this type of register, magnetic cores are among the most recent to become successful, and it is with them that the shortest access times are possible. Circuits in which the cores are employed are characterized by compactness, low power requirements, and long life with consistently reliable operation.

Switching Concepts. Magnetic cores suited to the application exhibit a rectangular hysteresis characteristic, illustrated in the idealized loop of Fig. 12.6. The positive residual state B_r is designated binary 1, the negative state $-B_r$ binary 0.

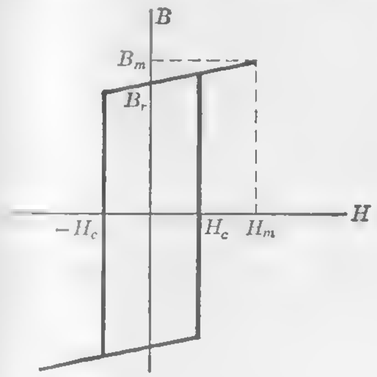


FIG. 12.6. Hysteresis loop.

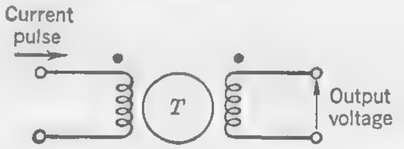


FIG. 12.7. Core transformer.

Application of a pulsed magnetizing force H_m of sufficient amplitude and duration will set the core in a desired binary state where it will remain until pulsed in the opposite sense. Reapplication of magnetization in the same sense results in relatively little flux change, e.g., from point B_r to B_m and back to B_r.

The core may be used as a controllable transformer (Fig. 12.7) with output coupled to the input of another core or device. Conventional dot notation indicates winding

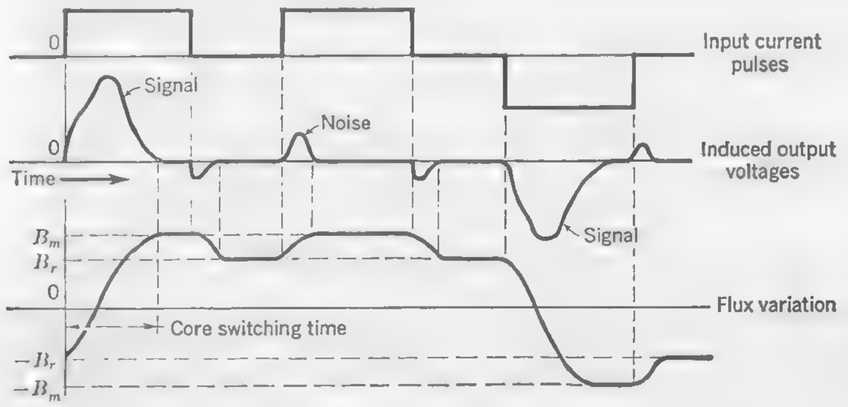


FIG. 12.8. Typical waveforms.

polarity with the added definition that current *into* a DOT terminal will set the core to the 0 state. Typical waveforms of voltages induced in the output winding are illustrated in Fig. 12.8 together with total flux variations within the core material. Signal-to-noise voltage ratios are generally greater than 20:1 with commercially available cores.

* By Isaac L. Auerbach.

The core may also be regarded as a variable impedance (Fig. 12.9). If the core is in the 1 state when I is applied, the core will switch, a relatively large counter emf e will be generated in the winding, and a high impedance will be presented to the driving source during the period of switching. If in the 0 state, the emf and presented impedance will be small. This bistable impedance characteristic is the basis for a number of circuits.

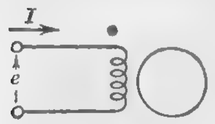


FIG. 12.9. Variable impedance.

Logical Representation. A simplified functional representation is helpful in understanding the operation of magnetic-core systems and as a basis for circuit design. A core is represented by a circle. A line with an arrow pointing to the circle (Fig. 12.10) denotes an input to the core; the numeral within the circle indicates the state to which the core is set by the input. Open arrows indicate pulses, closed arrows d-c signals. Two opposed signals applied simultaneously cancel, leaving the core in the original state. Double arrows indicate overriding inputs that determine the state of the core regardless of opposing inputs.

A line originating at the circle denotes an output from the core. It will later be seen that output signals are generally limited to a single polarity by the use of unilateral elements; thus the numeral within the circle at the origin of the line indicates the state *to which* the core must be switched from the other state to produce the output. If an output is produced by only one of several transfer inputs, the signal output is said to be conditional, indicated by an "eyebrow" drawn within the circle. Symbols along lines identify signals and often include the times at which the data appear.

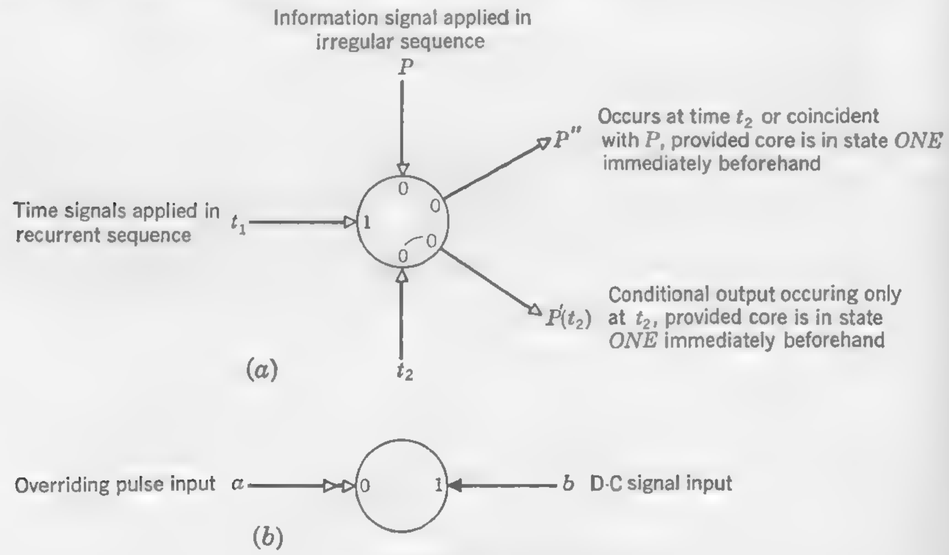


FIG. 12.10. Logical representation.

Fundamental Equations. Since the dependence of flux density B upon magnetic-field intensity H in magnetic materials cannot be reduced to a satisfactory equation, this relationship is ordinarily shown graphically, as in the generalized hysteresis characteristic where

$$B = \frac{\Phi}{A} \qquad \text{(MAGNETIC FLUX PER UNIT SECTIONAL AREA)} \qquad (12.3.1)$$

and $$H = \frac{F}{L} = \frac{NI}{L} \qquad \text{(AMPERE-TURNS PER UNIT LENGTH OF MAGNETIC PATH)} \qquad (12.3.2)$$

In the case of specific magnetic cores, where the physical dimensions are known or assumed, the derived relation of total flux Φ and applied magnetomotive force $F = NI$ is of practical importance. When several windings are involved, NI represents the

algebraic sum of the ampere-turns of the several windings. The symbol $F_0 = (NI)_0$, frequently encountered, is defined as the threshold force, or the value below which no switching is possible from a stable state.

The following relation holds during the switching of cores, in which e is the instantaneous voltage induced across a winding of N turns, and $d\Phi/dt$ the time rate of change of flux.

$$e = N\frac{d\Phi}{dt} \qquad (12.3.3)$$

$$E_{av} = N\frac{\varphi_1 - \varphi_2}{t} \times 10^{-2} \text{ volt} \qquad (12.3.4)$$

in which φ and t are expressed in maxwells and microseconds, respectively.

The switching time of a core is defined as the time required for a core to be driven from a residual state to the point B_s (Fig. 12.6) of opposite polarity. If F is plotted as a function of the reciprocal of the observed switching time for a given core, the curve is approximately linear for the limits considered and can be written

$$F = F_0 + \frac{G}{t} \qquad (12.3.5)$$

where G is a constant and intercept F_0 corresponds to an infinite switching time. If the applied NI reach a constant maximum value F after a time rise T_r, assuming a trapezoidal driving current, it has been found experimentally that the following relation holds, provided T_r is short with respect to t.

$$F = F_0 + \frac{G}{t - \frac{1}{2}T_r} \qquad (12.3.6)$$

Equation (12.3.5) is a particular case of the above where T_r is zero.

12.3.2. Transfer Loops. Transfer loops are networks connecting two or more cores for the purpose of information transfer. First suggested by Professor Howard Aiken of the Harvard Computation Laboratory in an early version of a magnetic shifting register, the original circuit arrangement has undergone a number of changes, resulting in types of improved performance.

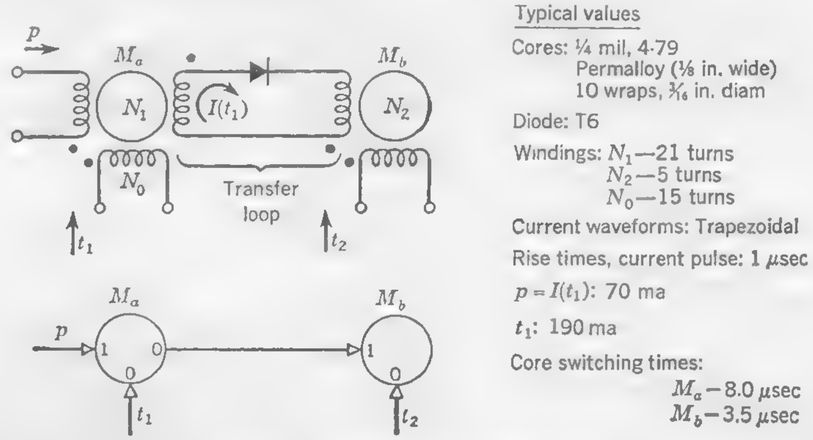
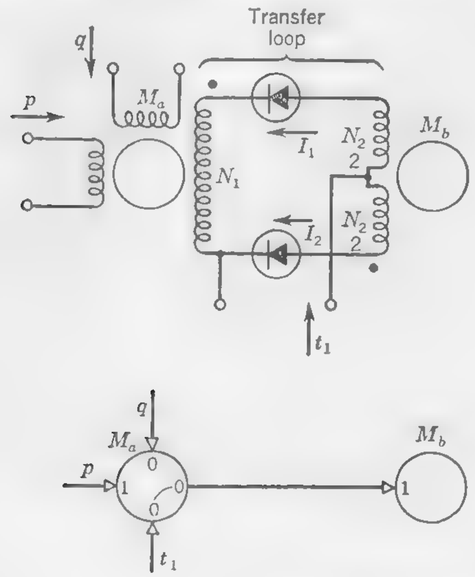


FIG. 12.11. Single-diode transfer loop.

Single-diode Transfer Loop. This is the simplest form of loop, illustrated schematically in Fig. 12.11, together with the logical representation. The diode isolates core M_b when a 1 is being set into M_a. A turns ratio N_1/N_2 of about 4:1 and the nonlinear characteristic of the diode, discriminating against the smaller signals, prevent an undesired backward flow of information at time t_2.

The use of this type of loop in logical circuits is limited, as additional windings similar to N_0 likewise effect transfer of information. In this instance the transfers are said to be unconditional.

Split-winding Transfer Loop. This type of loop, one form of which is shown in Fig. 12.12, provides for conditional transfer of information while permitting isolated



Typical values

Cores: ¼ mil, 4-79 Permalloy (⅛ in. wide) 10 wraps, ⅜ in. diam

Diodes: T6

Windings: N_1 — 31 turns
N_2 — 6 turns

Coil resistance, N_2—8 ohms

Current waveforms: Trapezoidal

Rise times, current pulse: 1 μsec

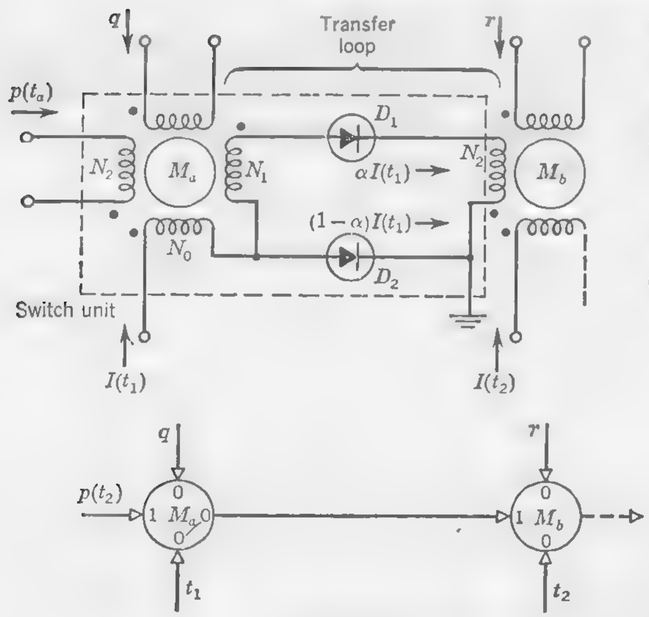
t_1: 125 ma

I_1: 8 ma

I_2: 117 ma

Core switching times:
M_a: 10.4 μsec
M_b: 4.0 μsec

FIG. 12.12. Split-winding transfer loop.



Typical values

Cores: Ferrite S3, General Ceramics

Diodes: T25G

Windings: N_0 — 23 turns
N_1 — 20 turns
N_2 — 8 turns

$I(t_1) = 100$ ma
$= 0.9$

Core switching times:
$M_a = 3$, $M_b = 2$ μsec

FIG. 12.13. Karnaugh switch.

operations on either the transmitting or the receiving core. Unlike the single-diode loop, it is immune to the backward flow of information, making practicable the simultaneous switching of as many as five or six receiving cores. By proper design the impedance of N_1 will be large relative to the other impedances in the transfer loop (cf. Fig. 12.9), and branch current I_1 will be smaller than I_2. The resultant ampere-

turns difference is sufficient to switch M_b to 1. Branch current I_1 clears M_a to the 0 state.

Karnaugh Magnetic Switch. The conditional transfer loop of Fig. 12.13 makes use of a basic arrangement known as the Karnaugh magnetic switch. As in the case of the split-winding loop, the two diodes insure immunity to the backward flow of information, permitting isolated operations upon either core. The function of conditional transfer from M_a to M_b in response to $I(t_1)$, while analogous to that of the preceding circuit, is accomplished in a somewhat different manner. In the properly designed circuit, current pulse $I(t_1)$ is steered through either the upper or the lower branch of the loop, depending upon the state of core M_a immediately prior to t_1. If in the 0 state, except for effects traceable to a nonideal core, all the applied current will flow through the high-conductance diode D_2, by-passing N_2 and leaving core M_b in the 0 state. When M_a previously has been set to 1, $I(t_1)$ acting through N_0 switches the transmitting core toward 0. The changing flux induces a voltage in the output loop such that nearly all the current flows through D_1 and N_2, switching M_b to 1. In practical circuits, parameters are generally so chosen that approximately 10 per cent of the current flows through the lower loop during transfer to stabilize the switching action of core M_a.*

Design Equations

$N_0 = \dfrac{NI_0}{\alpha I(t_1)}$ — Where NI_0 is the magnetomotive force in ampere-turns corresponding to the selected switching time of the driven core, obtained from constant-current response characteristics for the core

$N_1 = \dfrac{\alpha I(t_1) Z_L}{V_0}$ — Where V_0 is equal to the average volts per turn generated in winding N_1 during switching of core M_a and Z_L is the impedance of the load path

$N_1 = N_2 + \dfrac{V_d}{V_0}$ — Where V_d is the voltage drop across diode D_1. Applicable in the case of identical cores

$N_0 = \alpha N_1 + \dfrac{Ni(t) \text{ min}}{I(t_1)}$ — Where $Ni(t)$ min is the minimum ampere-turns required to maintain V_0, obtained from constant-voltage response characteristics

NOTE: To ensure a complete forward transfer of data, the switching time of the driving core must be longer than that of the driven core.

Condenser Delay. In certain data-handling systems it is often desirable to transfer information both *to* and *from* the same core by application of a single timing pulse. Such operation is possible if provision is made in the transfer loop to delay the input to the core until READ OUT has been accomplished. Assuming the cores of Fig. 12.14 contain 1's, both are switched to 0 at times t. The delayed output of M_a then resets M_b to 1 before arrival of the next timing pulse.

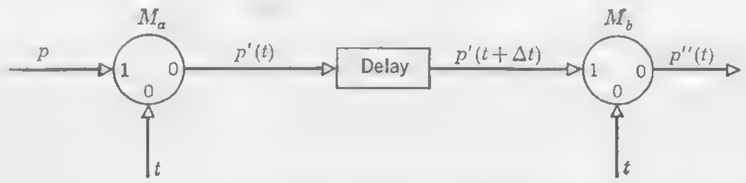


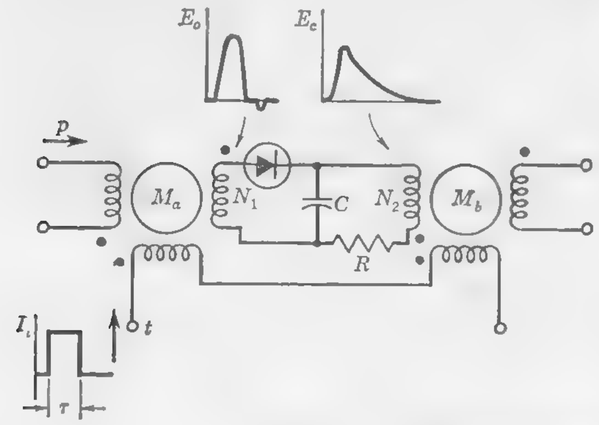
FIG. 12.14. Delay transfer.

Figure 12.15 illustrates a practical method for accomplishing the above.[42] If the forward resistance of the diode is small compared with R, the capacitor charges to nearly the peak value of the induced voltage as M_a is switched by t, and sufficient charge is present to switch M_b to 1 after termination of t. These loops have been operated at frequencies as high as 100 kc in cascades up to 50 stages with high stability, and provide added flexibility in the use of magnetic-core circuits.

* This means effectively that the current signal generated in the loop does not completely cancel $I(t_1)$ in the lower branch of the loop.

Transistor–Magnetic-core Loop. This is a magnetic-core transfer loop that includes a transistor as a power-supplying element. Generally a grounded-emitter configuration is employed with at least one core winding in the collector branch. Because of the element of gain, input signals need only trigger the transistor to effect switching of the cores, and possible speeds of operation exceed those attainable with passive loops. Power consumption is characteristically low, and wide tolerances in component values and transistor parameters ensure good reliability of operation.

A basic loop, applied successfully in circuits operating in the neighborhood of 100 to 150 kc,[24] is shown in Fig. 12.16. The transistor is biased at cutoff and is



Approximate circuit requirements

Core switching times:

$M_a \doteq \tau$

$M_b = 1.75RC$

Turns ratio:

$$\frac{N_1}{N_2} = \frac{\tau}{1.6\,ZC}(1+2m)$$

where Z equals average impedance of N_2 during read-in of *ONE*, and m is ratio of average drop across diode during charge to E_c (max)

Maximum operating frequency:

$$f_m = \frac{1}{\tau + 2.5\,RC}$$

Fig. 12.15. Condenser delay transfer.

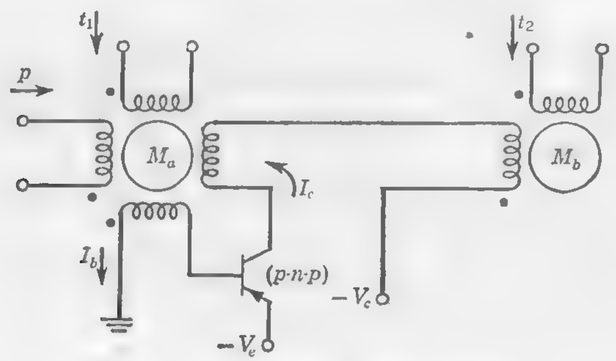


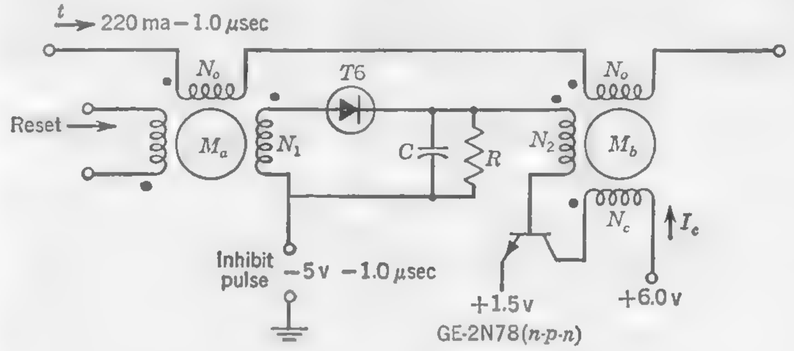
Fig. 12.16. Transistor transfer loop.

normally nonconducting. Because of coupling through the core a fractional-microsecond current pulse t_1 is sufficient to initiate a regenerative switching process, provided M_a contains a 1. When switching of the cores is completed the feedback-loop gain falls below unity and the transistor again ceases to conduct. Reset of M_a to 1 results in the appearance of a positive voltage at the base, tending to drive the transistor further into cutoff. The buffering action of the transistor eliminates backward flow of information in the loop (noise) and permits multiple branching between cores without difficulty.

The transistor-core loop of Fig. 12.17 illustrates an alternate arrangement of improved stability that employs condenser delay (cf. Fig. 12.15) to enable read-out and read-in of core M_b by means of a single timing pulse. Current pulse t switches both cores (if previously in state 1) to 0, while the time-coincident inhibit pulse maintains transistor cutoff. At the termination of the pulses, the voltage across the capacitor is sufficient to initiate switching of M_b to 1. Operational frequencies up to 500 kc are possible by the use of fast surface-barrier transistors and very low flux cores.

12.3.3. Noise Generation. The presence of spurious information signals, or noise, may be traced in some instances to properties of the cores themselves, or to characteristics of the circuits in which they are employed. In practical circuits, levels are limited, by proper design, to values that can be tolerated without sacrifice of operational stability.

Forward Noise in Single-diode Loops. One of the basic sources of noise in magnetic-core circuits is the nonsquareness of the hysteresis characteristic of the core. The amount of flux change produced when the core is driven from a residual state to the saturation point of the same polarity can be considered as false information.* The amount of energy involved may be small; however, the rate at which the flux changes can be high, resulting in peak voltage amplitudes approaching that of the true information pulse.



Circuit values — 200 kc operation

N_o — 5 turns	N_2 — 10 turns	C — 0.001 μf
N_1 — 15 turns	N_c — 30 turns	R — 4.7 K

Fig. 12.17. Transistor with condenser delay.

Back Noise in Single-diode Loops. In general, where loops provide for unidirectional transfer of information, induced voltages tending to effect transfer in an opposing sense must be considered as a potential noise source. The limiting of noise currents of this origin is of particular importance in the case of single-diode loops and has been included in the discussion of Art. 12.3.2.

Noise Due to Circuit Ringing. Distributed capacitances associated with core windings usually are small and may be neglected. However, dependent upon the number of turns, the shunt capacity may assume such values that energy storage during switching of a core and subsequent return to the circuit can result in partial reset of the core.

Transient Forward Noise in Split-winding Loops. In the case of 0 transfer where both cores of the basic circuit (Fig. 12.12) initially are in the 0 state when t_1 is applied, it has been assumed that the diode characteristics are identical, the coil resistance of N_2 is equally divided, and the impedance of coil N_1 is negligible. Under these conditions, the branch currents are equal and the net ampere-turns applied to M_b vanish.

However, nonsquareness of the hysteresis characteristic results in a small but undesirable inductance in N_1 and a transient inequality in the branch currents that tends to switch M_b to 1. This forward noise often is reduced by adding a balancing inductance to the lower branch.

Branch-resistance imbalance similarly may result in both transient and steady-state inequalities conducive to noise. Although undesirable from a power standpoint, branch resistances frequently are augmented by the use of resistance wire in the split winding as a means of minimizing noise levels.

* Refer to Figs. 12.6 and 12.8. The squareness of a core is defined by the ratio B_r/B_m, which approaches unity as the ideal value.

Potential Noise Sources in Systems. In many register systems,* the data are shifted along the core array by application of "advance" pulses which furnish the total energy required in the operation by the individual transfer loops. The number of advance lines is dependent upon the core configuration and coupling employed; however, each line usually consists of excitation windings of selected register stages joined

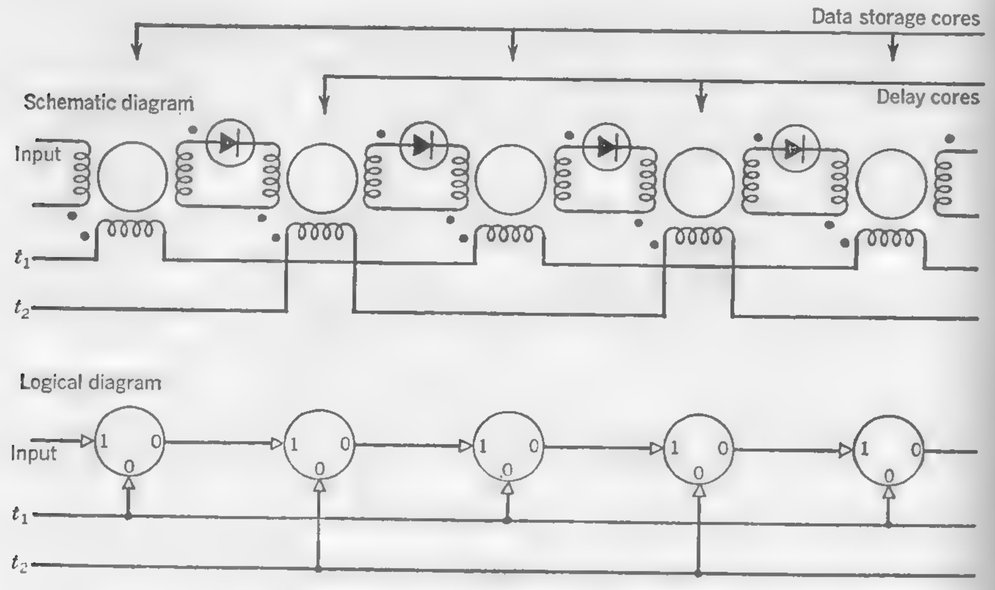


FIG. 12.18. Shift register, serial input and output.

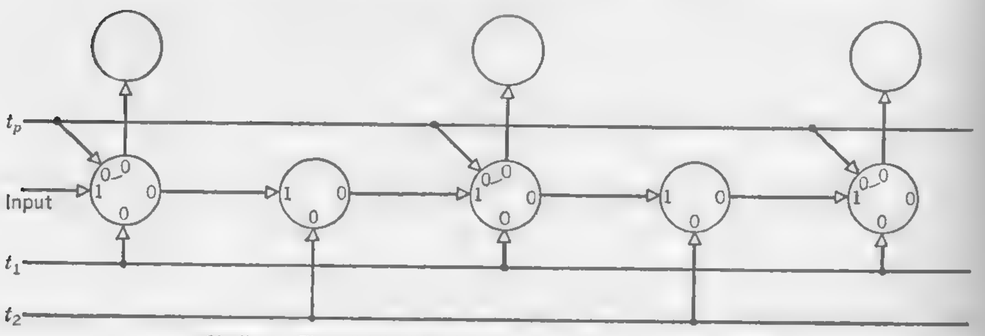

t_p
Input
t_1
t_2
Similar to Fig. 12.18 with addition of conditional transfer loops.
Parallel read-out accomplished by t_p applied as required at times other than t_1, t_2, or during input.

FIG. 12.19. Shift register with parallel read-out.

in series with a separate constant-current driver. When many cores are assigned a single advance line, the series inductance and resistance of the equivalent network during switching, together with the shunt wiring capacities, may result in circuit behavior typical of pulsed delay lines. In addition to a progressive delaying effect upon switching, the current rise time normally increases as the current passes through successive cores. This rise-time deterioration, when present, may cause erratic

* See Art. 12.3.5. Although not necessarily periodic, the advance pulses are referred to as timing pulses in the reference, principally for convenience in emphasizing normal sequence or time order of application.

operation toward the end of the register because of the increased effective resistance of the diodes* present in the coupling networks.

Upon the application of a rectangular or a very fast-rising current pulse to a core-driven† stage, leakage inductance associated with the magnetic coupling initially tends to isolate both the forward and reverse transfer loops from the transmitting core. As a consequence, the driving core tends to switch quickly, while the receiving core starts to switch quite slowly. If the leakage is appreciable, the transfer of data through the forward loop actually may be prevented. Increasing the rise time of the advance pulse reduces bucking voltages of this origin; however, the previously considered effects of increased diode resistance are apt to become troublesome. In practice, the rise time usually is chosen as a matter of compromise.

12.3.4. Typical Magnetic Shifting Registers. Shift registers for handling binary data are synthesized by means of magnetic cores and transfer loops. By suitable

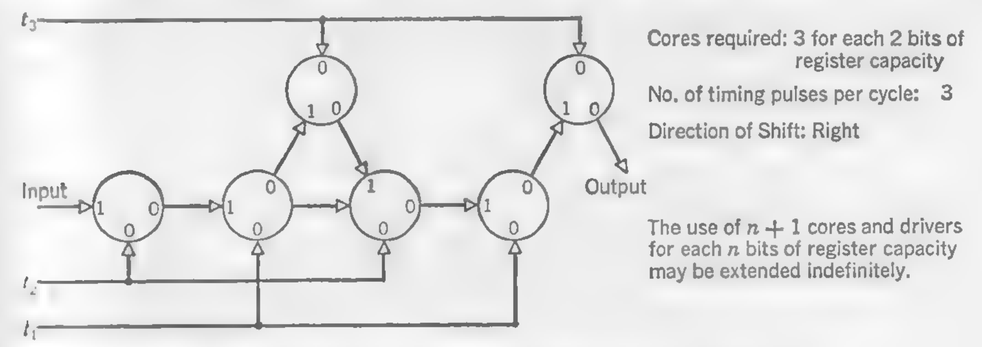

t_3
Input
Output
t_2
t_1
Cores required: 3 for each 2 bits of register capacity
No. of timing pulses per cycle: 3
Direction of Shift: Right
The use of $n + 1$ cores and drivers for each n bits of register capacity may be extended indefinitely.

FIG. 12.20. Serial shift register, reduced core complement.

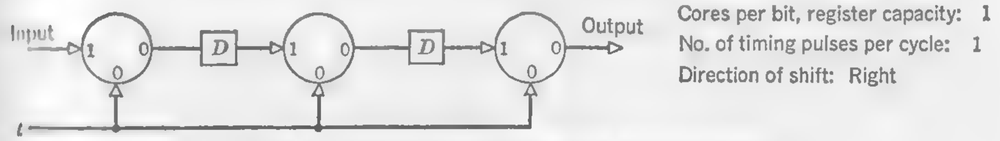


FIG. 12.21. Serial shift register with delay network.

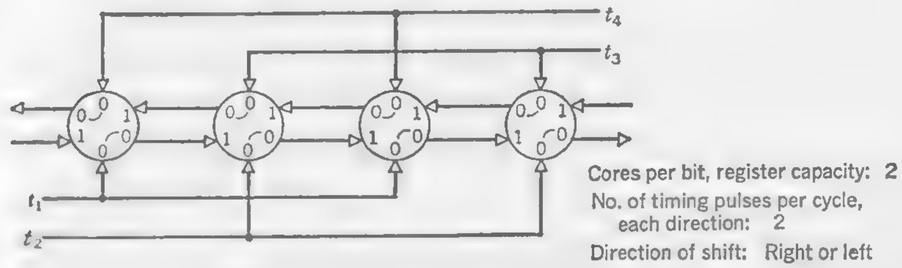


FIG. 12.22. Reversible serial shift register.

design and choice of coupling determined from preliminary logical diagrams, registers may be constructed that accept information serially or in parallel, step the recorded sequence along the register in either direction, and furnish a serial or parallel output. A complete timing cycle (t_1, . . . , t_n) accomplishes a shift by one binary place, for example, 100100 to 010010 in a 6-bit register.

Ping-pong Register. Since, in general, output delivery is accompanied by clearing of the register, means must be provided to restore the information to the register if

* See Diode Characteristics, Art. 12.3.6.

† Transfer loops, in general, may be classified as core-driven or directly driven types, depending upon whether a separate winding is specified for the advance function. The distinction may be seen by reference to the circuits of Figs. 12.11 and 12.12.

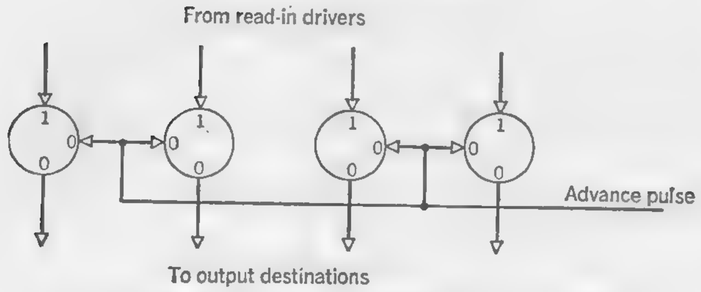


FIG. 12.23. Shift register, parallel input and output.

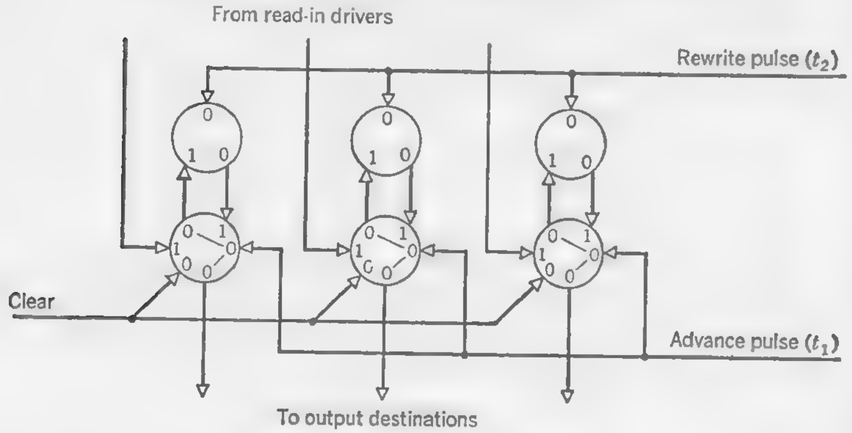


FIG. 12.24. Ping-pong register.

nondestructive read-out is required. This may be accomplished by the use of "ping-pong" circuits, as illustrated in Fig. 12.24, which is a modification of a 3-bit parallel register.

12.3.5. Magnetic-core Characteristics. Bistable magnetic cores have been developed in two basic forms, both of which are normally toroidal. The wound-tape type, consisting of wraps of microthin metallic strip, is usually employed in magnetic shifting registers, while molded ferrites are more often found in large-scale static memory applications. Marked differences between the two are summarized in Table 12.1. A variety of cores of both types, varying in diameter from ½ in. to 90 mils, are available.

Table 12.1

	Wound-tape	Ferrite
Construction	Insulated cold-rolled metal strip wound on ceramic bobbin and spot-welded in place. Heat-treated in dry hydrogen. Sometimes packaged	Mixture of finely ground iron oxides, pure metals, organic binder, pressed and fired at high temperatures in reducing atmosphere
Typical dimensions	Tape thickness 0.125 to 1 mil Tape width 0.125 to 0.25 in. Core diam 0.125 to 1.00 in.	Outside diam 0.050 in. Inside diam 0.030 in. Thickness 0.020 in.
Resistivity	.....	High
Flux density	Medium to high	Low
Coercivity	Small	Large
Appropriate application	Manipulative-type circuitry requiring numerous turns of copper wire in windings	Magnetic memories where windings may consist of single wire through center

Characteristic Curves. The d-c hysteresis loop (cf. Fig. 12.6), with which the operating characteristics of a core are most easily interpreted, is obtained by varying the current in a winding slowly so that, for each instantaneous value of current, the magnetic field within the core material has sufficient time to reach an equilibrium value. If the variations of current are very rapid, as under pulse conditions, the values of B show an additional dependence in which time is involved.

From experimental results obtained in high-speed operation, it appears that complete switching is possible even though complete saturation is not reached and that boundary lines of the d-c loop are replaced by boundary zones. However, the widths of the zones are small enough to justify general reference to the d-c loop.

Some experimental curves applicable to metallic strip of specified material, thickness, and width are shown in Fig. 12.25. The tolerance limits of switching time, median flux change from one stable state to the other, and squareness ratio B_r/B_m are plotted against magnetic-field intensity H. The latter two parameters, constant by d-c loop interpretation, exhibit slight variations in keeping with the high-speed switching concept. Maximum and minimum values of G and F_0, defined in Art. 12.3.1, are indicated as simple functions of mean core diameter.

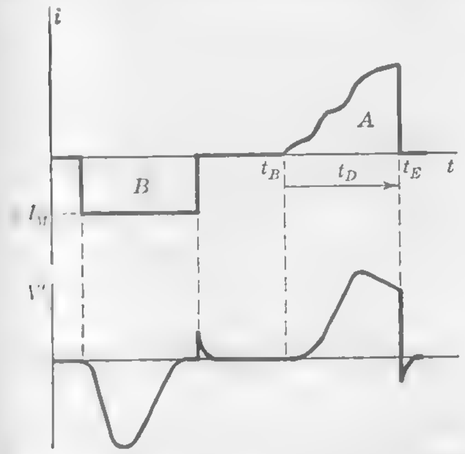


FIG. 12.25. Experimental curves for metallic material.

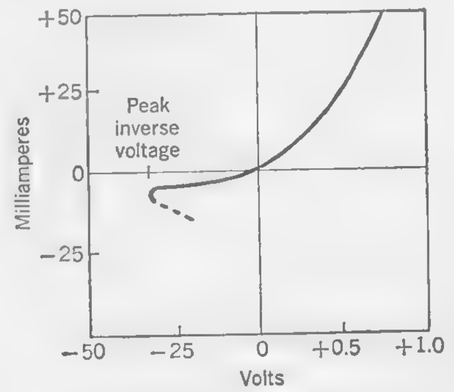


FIG. 12.26. D-C characteristic.

12.3.6. Diode Characteristics. The use of series diodes has been indicated in discussion of the various transfer loops associated with magnetic shifting registers. In general, commercially available germanium crystals prove to be satisfactory, provided due consideration is given manufacturers' ratings and operating characteristics for the type selected.

Steady-state Characteristics. A typical d-c characteristic is shown in Fig. 12.26. The theory of the crystal rectifier is not yet complete, but it may be seen that the rise of current approximates a positive exponential rising from a negative current value and passing through the origin. The deviations that are always present in physical units may be represented by a resistance in series with a truly exponential element.

In the range of inverse voltages encountered in the loop circuits (of the order of −30 volt), the back resistance varies considerably with temperature and humidity. The resistance value generally is unimportant, provided it exceeds a few thousand ohms. The forward resistance, of more critical importance, is relatively insensitive to ambient factors.

Transient Forward Characteristics. Figure 12.27 illustrates a typical diode forward voltage developed by application of current pulses of fractional-microsecond rise time. During current rise, the resistance of the diode, characteristically high to small currents, is augmented by the limited ability of the semiconductor to produce holes at an accelerated rate. As a result, the voltage curve exhibits an initial peak and gradual reduction to the steady-state value.

The average diode forward voltage, frequently employed in design computation, may be approximated from the presentation.*

Recovery Times. Although often regarded as negligible for the types of circuits and components being considered, an additional transient effect is of interest in a general discussion of crystal diodes. If the time between the termination of a forward current pulse and the rise of an inverse voltage is zero, a small but finite interval, referred to as *active* recovery time, is necessary for the diode to begin development of back resistance.

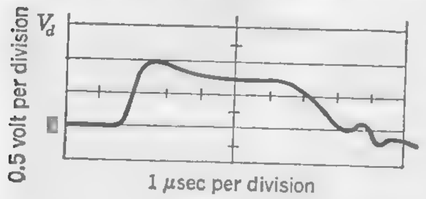


FIG. 12.27. Typical diode forward voltage.

The minimum delay that must follow the termination of the forward pulse for coincident initiation of inverse voltage rise and resistance development is known as *passive* recovery time.

12.3.7. Drivers for Magnetic-core Circuits. While the introduction of bistable magnetic cores has led to considerable advances in reliability and miniaturization of data-processing systems, more conventional elements and techniques are still often to be found in auxiliary power equipment. Except for the more recently developed transfer loops containing active elements,† pulse power from an external source is required to drive the magnetic networks and is commonly derived from electron-tube-type generators. Vacuum and gas tubes, transistors, and magnetic amplifiers all play a part in the continuing development and improvement of drivers; however, it is upon the latter two that the greatest interest is focused. The description of this type of circuitry, in detail, lies outside the scope of the present chapter.

Waveforms most commonly suited to the application are most easily obtained from constant-current sources. The pulses sometimes may be nearly rectangular but more often are approximated by trapezoidal functions. Stability in the output is, of course, highly desirable—particularly with respect to amplitude over the range of impedances presented by the variable load conditions.

Vacuum-tube pentodes triggered by appropriate pulses, blocking-oscillator-type circuits, and self-extinguishing thyratron circuits serve effectively as drivers for magnetic-core circuits. The latter two can be triggered by the output from a single core and may be used as amplifiers where it is required that one transmitting core switch a large number of receiving cores. Frequencies of 500 kc, more than adequate for the efficient operation of many present systems, are obtainable without difficulty, except for the thyratron circuits which are relatively limited in this respect.

The transistor, especially the junction type, is an obvious replacement for the electron tube in the continuing trend toward reduction of size and power consumption of data-handling systems. Among the circuits undergoing development in many quarters, a two-transistor regenerative‡ driver has shown considerable promise in this direction. This experimental driver provides 1-μsec pulses of up to ½ amp amplitude—an output sufficient for driving fifteen 5-wrap 0.033-microweber cores.

12.4. MERCURY AND QUARTZ DELAY LINES

The velocity of sound in mercury is about 5 ft per msec. A number of computers were built in both this country and England using mercury for the memory. In some cases the mercury was placed in steel or glass tubes with quartz-crystal transducers on each end. A carrier frequency of from 7 to 30 Mc was used and the information was stored in the lines as a modulation pattern on the carrier. The circuits

* Average voltage over the interval τ is defined as $(1/\tau)\int_0^\tau v_d\,dt$, where v_d is the instantaneous diode voltage.

† The transistor bistable magnetic-core combination of Fig. 12.17 is a practical example of this type of network.

‡ The regenerative principle employed in this driver is that applicable to the circuit of Fig. 12.17.

were arranged so that the information recirculated and hence given information was available on a cyclic basis. Magnetic cores are replacing mercury memory systems in this country in new equipment. In applications where delay storage is still used quartz has replaced the mercury. Typical characteristics of such a quartz memory line are:

Delay: 20 to 1,100 μsec with accuracy of 0.1 μsec
Input:
  Information: 8-volt positive pulses at 1 Mc
  Clock input: 10-volt positive pulses 0.5 μsec wide at 1 Mc
Output: 12-volt positive pulses 0.4 μsec wide

Such a unit may be about ⅓ cu ft in volume and use less than 60 watts of power. Half of that power may be used to control the temperature of the quartz line in order to give precise delays.

The above characteristics are obtainable only with a relatively complex circuit, since there is substantial loss in the quartz memory system. An example of such a circuit is shown in Fig. 12.28. This circuit uses four 6AN5 tubes. Long delay paths may be obtained by reflection inside quartz blocks (see Fig. 12.29). The intersection of such paths gives no difficulty.

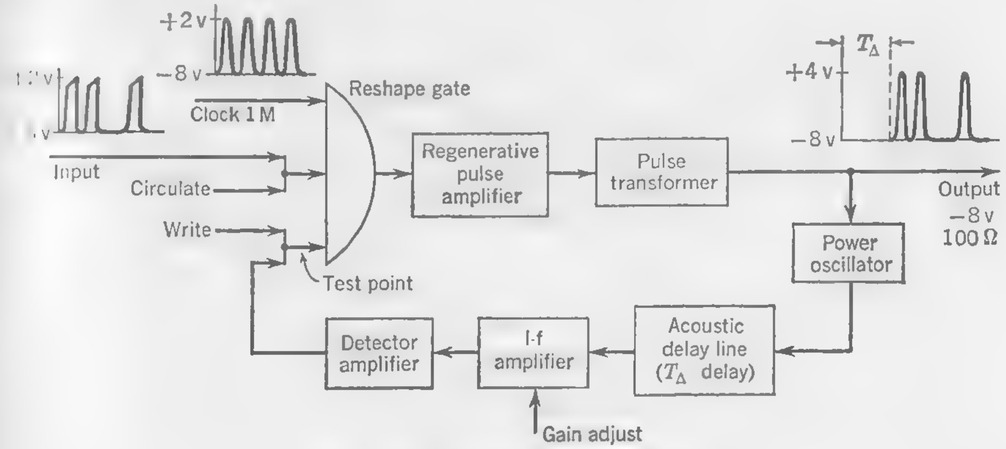


FIG. 12.28. Acoustic memory system.

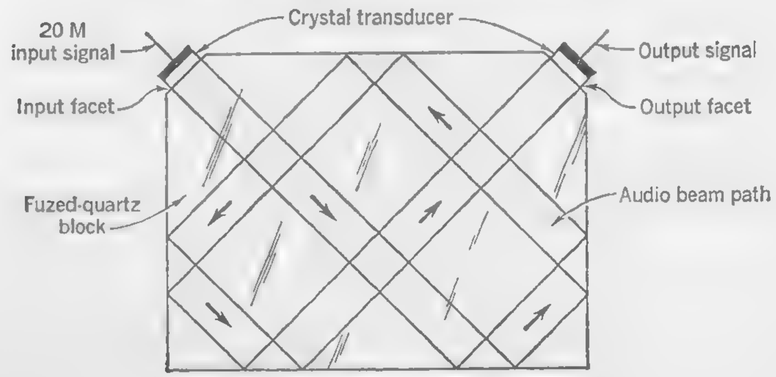


FIG. 12.29. Quartz acoustic memory.

12.5. MAGNETIC MEMORY SYSTEMS*

Magnetic recording as a method of providing the function of memory has attained a major position in the field of digital data storage, appearing to date most extensively

* By Albert S. Hoagland.

in the form of magnetic-drum, magnetic-tape, and magnetic-disk memory units. An entire class of electronic calculators, the so-called intermediate-speed computers, are designed around the magnetic drum as their working memory; and these machines represent the bulk of the electronic computers in existence today. Magnetic tape and disk devices are exclusively used for on-line *mass storage*. In the file-maintenance operations of data processing magnetic tape is now a vital systems component, although random-access disk storage is assuming an increasingly significant role. This later trend reflects a growing awareness that the potential of technological advances in magnetic recording offers a means to maintain effectively up-to-date records and to achieve current status control in business operations. New advances based on magnetic recording as a storage mechanism will tremendously extend the range of application and degree of utilization of magnetic data recording. Reasons for this wide exploitation of magnetic recording are based upon its reusability, flexibility of mechanical structure, economy in terms of cost per bit, compactness for large capacities, and ruggedness. The advantages of magnetic recording for data storage stem from the high storage densities obtainable, the simplicity of the components necessary, and the use of mechanical motion to reduce effectively the complexity in access at the cost of time.

Although it is only in recent years that magnetic recording has come into general use, its invention by the Danish engineer Valdemar Poulsen dates back to 1898. The first apparatus used a steel wire for the recording medium, while today a thin magnetic layer is used exclusively for this purpose in the computer field. The first application for this process was the recording and reproduction of sound. However, it was not until 1941, when significant improvements in components and recording techniques were combined, that this art began to receive the attention that was to result in its rapidly expanding growth, which is still continuing.

The actual work which marked the beginning of the application of magnetic recording to data storage was undertaken in 1947. The first practical units were magnetic drums. In common with all semipermanent memory devices for digital data, a bistable phenomenon is exploited. Thus, information is stored in binary form, and for this purpose two opposite senses of saturation of the recording medium are employed. In special cases where a pre-erase before write operation is practical, it is feasible to consider using two opposite senses of magnetization which may not be those of saturation. However, the practice to date is all predicated upon saturation of the recording medium. Thus, for data recording the signal current assumes only two values; the writing current may consist of either positive or negative saturation-current pulses, or levels. Read-back of such information essentially involves only the discrimination between two values of some attribute of the recovered waveform.

The various types of units in which magnetic recording provides the data storage differ between themselves essentially only in the mechanical structure used to provide relative motion between the surface and from one to many magnetic transducers. These devices will be described in some detail later, but attention is now called to the most common arrangements employed.

A rotating drum whose surface possesses a magnetic-recording film serves the function of a high-speed memory for intermediate-sized computers. Magnetic heads are located along the axis of the drum, slightly out of contact with the surface. Drum speeds are in the range of thousands of rpm, and in the customary size of drum, hundreds of thousands of binary digits can be stored with an access time to any bit in the order of milliseconds. Another application for which magnetic recording is well suited is that of providing recirculating storage loops with capacities ranging from several tens of bits to several hundred bits of information. This type of memory is a special form of drum usage where a surface location passes first a write station and then a read station, from which the information is regenerated and rerecorded at the write station. Depending upon the form of recording, there may be an erase station following the reading point. Extreme flexibility is available here; the length of the storage loop may be varied by simply moving the location of the reading station. The benefits from the elimination of attenuation problems, common to delay-line

devices, for this type of storage are obvious. This magnetic-recording application forms the basis for a class of machines known as electronic digital differential analyzers.

For high-speed computers it is desirable to provide a large-capacity buffer memory to back up the high-speed working memory. In this area magnetic-tape units are without peer. By providing several tape units and the facility for changing tape reels, the computer has at its disposal an extremely large intermediate-speed memory from which blocks of data may be transferred to the short-access memory when needed. In processing of business record files tape units provide the link to the computer proper. Further, the magnetic-tape units provide a means to match the computer proper to the terminal input-output equipment, whose speeds are orders of magnitude slower. Thus magnetic-tape units are found with nearly every high-speed computer and to find as many as twenty such units with an installation is not unusual. The reels of tape are in the neighborhood of 2,000 ft in length, and data organization is therefore essential.

In many data-processing applications, particularly those in business, a very large volume of information must be available on short notice. This need has placed emphasis on the concept of a large-capacity random-access memory. Two different magnetic devices, announced in 1956, are designed to fill this function. One technique uses short sections of tape in a binlike arrangement with mechanical selection of the tape strip. The other basic concept is similar to that of a jukebox. Here an array of magnetic disks provides the memory while the random-access feature is incorporated through mechanical positioning of the magnetic-head unit. The latter device is capable of a capacity of hundreds of millions of bits with an access time to any bit, or group of bits, of less than 1 sec.

Variations in storage possibilities with magnetic recording result from having two or more magnetic heads servicing the same track. A recirculating loop is obtained simply by assigning a write function to the leading magnetic head and using the trailing unit for read-back. The length of this delay line is established by the physical displacement between these transducers. Regeneration of the recorded information coupled with selective alteration provides a means to realize a small-capacity short-access storage on one to several of the recording tracks. As mentioned previously, the delay-line concept of magnetic recording is the heart of the electronic digital differential analyzer.

Important aspects of magnetic recording with respect to its use for data storage are the following: saturation recording is used and, therefore, new information may be written over that previously contained, erasing being unnecessary; the memory is nonvolatile, i.e, the information is not lost in the event of a power failure; there is no apparent deterioration of the recorded information with time; and the storage film can tolerate considerable physical abuse with no deleterious effects and is insensitive to wide variations in environmental conditions.

The advantages of these features of the magnetic-recording process contribute enormously to its attractiveness as a storage mechanism.

12.5.1. The Magnetic-recording Process. Magnetic recording, using the conventional magnetic transducer, relies on relative motion between a permanent magnetic-storage surface and a magnetic head. The magnetic head couples magnetically with a small region of the surface adjacent to it, magnetizing the surface on writing and providing an induced voltage due to the time variation of this surface magnetization on read-back. The path on the surface delimited by the read-write operations, because of relative motion between a transducer and the storage layer, is a recording track or simply a track. Recording is thus time-sequential with respect to a track. Surface storage is achieved by recording tracks parallel to one another; all these tracks are likewise parallel to the direction of relative motion. Mechanical considerations have resulted in the relative motion being provided almost exclusively by a moving surface and a stationary magnetic head.

The bit density per unit surface area is composed of the bit density per unit track length times the track density per unit distance normal to the direction of relative motion. Cost considerations stress the importance of exploiting bit density relative to track density to realize a given capacity, since this approach minimizes the number

of magnetic heads and the complexity of the mechanical assembly. Representative values for these recording densities are 200 bits per inch and 20 tracks per inch, corresponding to 4,000 bits of memory for every square inch of recording-surface area. Wide extremes are found in both these density figures.

A memory unit in relation to its external environment may be described in terms of capacity (number of binary digits that can be stored), mean access time (average waiting time for a given memory location to be accessible to its associated read-write station), and bit or data rate (scanning rate of a given transducer in bits per second). These factors are interrelated. For a given recording track, let bpi stand for storage density in bits per inch and let the relative velocity of the storage surface with respect to the magnetic head be designated as v. Then

$$\text{bpi} \times v = \text{bps} \tag{12.5.1}$$

where bps = bits per second flow rate. The access time bears an inverse relation to v. Further, bpi is directly proportional to memory capacity. The relation existing between these quantities is important, since it accounts for two fundamentally different types of mechanical-recording arrangements that are found in practice. These two categories may be essentially defined by the terms "contact recording" and "noncontact recording." In contact recording it is desired to achieve as intimate a physical (and hence magnetic) contact between the recording surface and the magnetic head as possible. For lack of a better definition, contact operation is considered here to imply that there is a finite rate of wear of one or both of the contacting surfaces. Contact operation offers the highest recording resolution or density potential. All audio and video magnetic-tape recorders use contact recording to obtain the broadest wavelength spectrum. However, to keep wear down to an acceptable rate in contact recording the maximum surface speed is severely limited. In data recording, where there is a premium on reliability, maintenance considerations are very important. At present, contact recording of binary information is normally limited to velocities in the neighborhood of 100 in. per sec in order to obtain satisfactory life characteristics. This limitation on surface velocity, likewise limits the available bit rate. A working memory should match the speed of the processing unit in data flow if the memory is not to handicap machine performance severely. The limitation of a relatively low bit rate with contact operation may be overcome in magnetic recording by providing a slight separation or spacing between the magnetic heads and the recording surface. This type of arrangement results in noncontact magnetic recording and allows an enormous extension in the range of applications of magnetic recording for data storage. The attendent loss in resolution with small values of spacing is not so severe as to be damaging, and the possible increase in relative motion (or surface speed) is in orders of magnitude, being limited only by the mechanical-drive problem. The net result is that a greatly increased bit rate is possible. For comparative purposes, consider contact recording along a track at 200 bits per inch (bpi) and a surface speed of 100 in. per sec contrasted with noncontact recording at 100 bpi with a surface speed of 1,000 in. per sec. The bit rate for the contact recording case is 20,000 bps while for the noncontact situation the bit rate is 100,000 bps. For primarily large-capacity storage it is advantageous to utilize the additional capacity afforded by contact operation; hence, magnetic-tape units use contact operation if at all feasible.

A fairly consistent terminology has developed with regard to the organization of data in magnetic-recording devices. Each magnetic head records on a track. Addressable locations around the circumference of rotating memory devices are called sections. All tracks simultaneously utilized in a parallel mode of data transfer, taken as a unit, may be referred to as a channel. A track recorded to provide timing information is referred to as a clock track. A recirculating magnetic loop, as described earlier, is commonly called a revolver.

12.5.2. The Magnetic-recording Theory. Magnetic recording involves writing information on a suitable recording medium and reading back the information so written. In writing, the magnetic head is energized by means of a coil and a suitable signal current, providing a sufficient magnetomotive force to produce an intense, local-

ized, magnetic field at the storage surface. Each succeeding region of the track is brought into a definite magnetized state as it passes by the magnetic head. The surface retains or "remembers" essentially this magnetic state because of its permanent magnetic character. This memory feature of the surface arises from the phenomenon of remanence, which is exhibited by ferromagnetic materials of which the surface is constituted. The recorded pattern of remanent magnetization along the storage surface is then related to the time variation of signal current. The connection between time and space is established by the surface velocity.

The external magnetic field along the surface, resulting from its remanent magnetization, couples magnetically with a reading head, which is almost always the same magnetic head used for writing. An output voltage is developed across a coil on the magnetic head because of the time variation of flux linkages.

In the writing and reading of digital (binary) information, this output voltage need not be a replica of the writing-current signal. It is sufficient that the value of a stored bit (0 or 1) be reliably recognized at any given location on the track. The over-all input-output storage-transfer process may be indicated in the following manner (neglecting frequency effects):

$$i(t) \rightarrow M(x) - \phi_h(\bar{x}) \rightarrow \frac{vNd\phi_h}{dx} = e(vt)$$

where $i(t)$ = writing-signal current
$\phi_h(x)$ = reading-coil flux as a function of the surface location relative to the magnetic head
e = open-circuit read-back output voltage
N = number of turns on the reading coil
x = surface coordinate
$\bar{x}$ = coordinate indicating the position of $M(x)$ relative to the magnetic head

Note that ϕ_h is the average flux per turn.

With the conventional induction-type transducer the fundamental transfer property is a derivative-type relation between input current and output voltage. Hence, when two senses of surface saturation are used exclusively, pulselike signals result on read-back, corresponding to the changes in writing current from one sense of saturation to the other. The output signal is inherently alternating in character. This basic nature of the read-back waveform may be obscured if the over-all system is operating under conditions where a change in current effects the change in magnetization produced by the previous switching of the current, or if time-dependent effects are important.

The magnetic properties desired for the recording surface are quite different from those necessary for the magnetic head. Consider first the recording layer. The value of the maximum-energy product for the surface medium $(BH)_{max}$ should be large. H is the magnetic-field intensity, while B is the induction-field intensity. Now the product $(BH)_{max}$ is an indication of the energy which may be stored in a unit volume of material. The induced voltage in the magnetic head on read-back is proportional to B_r, the residual induction, and hence a large value of B_r is desirable. For writing definition, a large coercive force H_c is beneficial (compatible with the means available for magnetization of the surface). This property enables a given region of the medium to be magnetized with less influence upon adjacent surface regions. A relatively large ratio of H_c/B_r is desirable for stability, as this will tend to counteract the influence of stray and self-demagnetizing fields.

The magnetic properties of magnetic-head materials should be similar in many respects to those of communication-type transformers. The initial permeability should be large for reading, this process involving weak fields. From the standpoint of writing they require a high saturation flux density in order to assure that the recording surface will always saturate first. The residual induction should be small so that the magnetic head is essentially in a demagnetized state when not energized.

In mks units $B = \mu(H + M)$, where M is the magnetic moment per unit volume of the material being considered. In the emu system

$$B = H + 4\pi M \tag{12.5.2}$$

where M is called the intensity of magnetization. Since the magnetic properties of materials are most commonly given in the latter system of units this system will be used. Here, B is measured in gauss and H in oersteds. When

$$M = kH \tag{12.5.3}$$

over a range of values of H(k = susceptibility), the expression for B may be modified to

$$B = B_0 + \mu H \tag{12.5.4}$$

or

$$\Delta B = \mu \, \Delta H \tag{12.5.5}$$

The last expression defines the differential permeability of the medium. This permeability is regularly used in describing magnetic materials, and henceforth the symbol μ or the word "permeability" will be used in this context.

The surface is a "hard" or high-coercive-force material while the head material is classified as a "soft" or low-coercive-force magnetic material. For the latter type of material the relation between B and H is reasonably linear over a wide range, and these factors are related by a very large value for μ. For hard materials the permeability near saturation is of the order of 1.0. These distinctive magnetic properties of the magnetic materials used permit a unique and in some ways qualitatively simple model of the magnetic-recording process.

The most common recording surfaces are made of magnetite (Fe_3O_4, black iron oxide), the synthetic red iron oxide (δ-Fe_2O_3), or an electrodeposited cobalt-nickel plating. These oxides are available in suspensions with an average particle size less than 1 micron in diameter. Typical magnetic characteristics for the 3M (Minnesota Mining & Manufacturing) red oxide suspension are $H_c = 250$ oersteds, $B_r = 500$ gauss. The base surface may be coated with this material by spraying, dipping, etc., after mixing with appropriate solvents. Coating thicknesses range from about 0.4 to 3 mils. For cobalt-nickel-plated surfaces H_c is around 200 oersteds and B_r is of the order of 10,000 gauss. However, their greater self-demagnetization (from a lower ratio of H_c/B_r) and possible eddy-current effects (from their high conductivity) tend to limit the thickness of a plated surface if comparable resolution is to be obtained. Fortunately, very thin films can be plated with excellent uniformity. Metallic tapes with plated surfaces less than 100 μin. in thickness are used. However, as indicated in the design section, the output voltage is roughly proportional to surface thickness, and hence the potential increase in output offered by the cobalt-nickel material cannot be realized.

The magnetic oxides have proved to be the most satisfactory surfaces to date. The one advantage of a plated surface is in the fact that it is physically harder than the oxide layers that have been developed; thus the plated surface is more resistant to accidental impact. One of the major problems in producing satisfactory oxide tape is to overcome oxide build-up resulting from the constant abrasion of the surface in contact operation.

Alloys such as 4-79 Permalloy, Hymu 80, and Mumetal are representative of laminated metallic-head materials. For high-frequency response thin laminations are needed and the thickness normally ranges from 2 to 7 mils.

Ferrites are achieving an increasingly important position as a head material. A laminated structure is not required because of their enormous resistivity. Ferrite is much less subject to wear but suffers from the fact that it is a ceramic material and cannot now be finished to provide as well-defined surfaces and edges as metallic alloys. The saturation flux density and permeability of the available ferrites are far lower than the values obtained with high-permeability metallic magnetic materials.

12.5.3. Magnetic-recording Theory; Head Design. While transducers based on flux-sensitive means are used for magnetic recording, almost all transducers in use today are conventional magnetic heads. With respect to the direction of relative motion between the magnetic head and the recording surface there are three primary modes of recording that can be defined, based upon the principal direction of saturation. These are:

1. Longitudinal or horizontal recording
2. Perpendicular or vertical recording
3. Transverse recording

Longitudinal recording refers to recording in which the principal magnetic-field component of surface magnetization is in the plane of the surface and parallel to the direction of surface motion. Occasionally this type of recording is called horizontal recording. In vertical recording the principal orientation of the writing field is normal to the plane of the surface. In transverse recording, the medium would be magnetized in the plane of its surface but normal to the direction of surface motion. These three modes of recording and the magnetic-head structures to produce them are illustrated in Fig. 12.30. The almost universal mode of recording today is longitudinal, using a ring-type magnetic head as shown in Fig. 12.30. The sketches indicate noncontact operation merely for the sake of clarity.

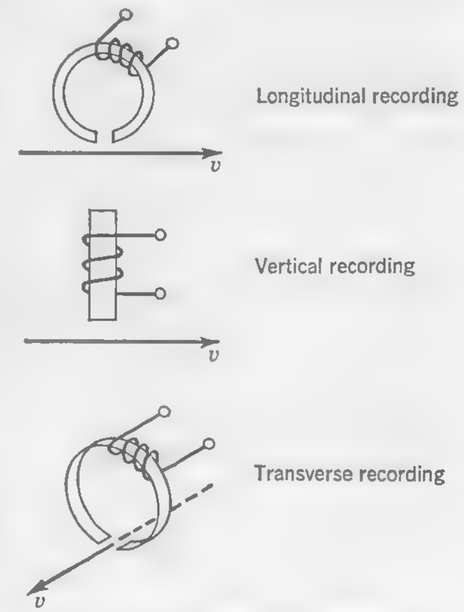


FIG. 12.30. Modes of recording.

The objectives in magnetic-head design are the same irrespective of the mode of recording; hence only design features pertinent to a ring-type head structure will be considered in any detail. The discussion will reveal several inherent advantages of this device. The ring-type magnetic head essentially comprises a low-reluctance magnetic path having a small nonmagnetic gap which is located next to the surface with the gap aligned normal to the direction of surface motion. This structure permits an intense localized magnetic field to be produced at the surface when a coil on this ring is energized. The writing field is actually the fringing field of the gap. On reading, this head effectively shunts the surface flux (due to the remanent surface magnetization), which otherwise would traverse the gap region, through the magnetic-head path. This flux thereby links a coil on this head and so induces a voltage proportional to the time rate of flux change.

For writing a coil must be able to provide a positive and negative magnetomotive force sufficient to develop a saturating-field intensity through the recording film in the region in proximity to the gap. In noncontact operation, this condition may require a magnetomotive force (mmf) in the neighborhood of 25 ampere-turns. It may be stated that for optimum recording performance the gap size should be nominally equal to the head-to-surface spacing in noncontact operation. This point will be clarified somewhat in the remainder of this subsection. In contact recording directed toward the highest possible storage density the gap size should be as small as possible, consistent with recording surface thickness. Fabrication problems have limited the production of contact heads to gap dimensions of a few tenths of a mil to date. The width of the magnetic head in the direction normal to the direction of surface motion establishes the width of the recording track, which is effectively equal to the dimension of the magnetic head. The track width should be small if a high track density, and thereby a high density per unit surface area, is to be achieved. However, the reading-output signal is proportional to track width. Further, the greater the track width, the less significant are small surface irregularities on the read-back-voltage signal. Head widths generally fall in a range from 7 to 100 mils (1 mil is equal to one-thousandth of an inch). Track densities correspondingly are of the order of 20 per inch.

The design of the magnetic head for high definition on writing and high resolution on reading is strongly influenced by the thickness of the recording medium, and in the

case of noncontact operation by the spacing between the magnetic head and the storage layer. Contact recording is merely a special case of noncontact recording in which the spacing has been reduced to zero in so far as parameter relationships are concerned. Hence writing definition refers to the size of the track region over which a saturation change in magnetization can be effected. This factor may also be defined in terms of the length along the track occupied by the magnetic "dipole," produced by a positive current impulse, applied to a negatively saturated track. The better the writing definition, the shorter the surface magnetization transitions. Reading resolution refers to the zone extending on either side of the magnetic head within which a steplike change in recorded magnetization causes a perceptible signal voltage (with the surface in motion). The playback signal obtained will be reduced in pulse width as the reading resolution is increased. For a magnetic head serving for both writing and reading, the nonlinear magnetic property of the surface—necessary for storage—results in an appreciably higher writing definition than reading resolution. Since reading and writing should be compatible when using a single magnetic head for both these functions, the design of the structure should be directed primarily toward improving reading resolution wherever opposing considerations arise.

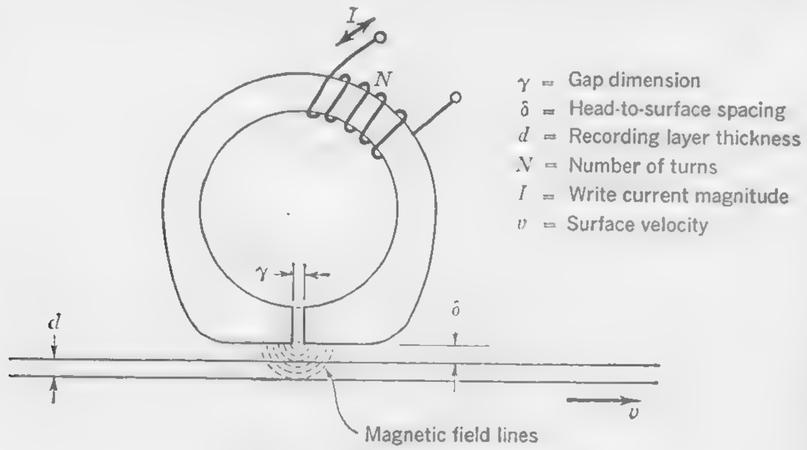


FIG. 12.31. Ring-head structure.

The ring-type head with several parameters indicated is shown in Fig. 12.31. The fringing field of the gap is the writing field that magnetizes the surface in the desired sequence. The gap size, saturation flux density, and writing mmf of the magnetic head must be chosen to produce a saturation writing field throughout the surface layer cross section immediately underneath the head gap. Since the fringing-field magnitude decreases at a fixed spacing as the gap size is decreased, the writing requirement will set a lower limit to gap size. However, as it is desired to cause the change in magnetization to occur over the shortest possible interval along the surface, the gap size should be no larger than necessary. A typical relation as mentioned previously is to set the gap at about the value of the effective spacing distance plus one half the surface-layer thickness. The surface layer should be as thin as possible, for writing must occur throughout this depth. The lower sections, of course, are at effectively larger spacings from the head than the top section. Though the fringing field is fairly broad in extent, its actual influence on the recording surface extends only over the distance wherein the saturation threshold field is exceeded. This distance is severely limited by the hysteresis phenomena exhibited by the surface. Since the surface passes by the gap, the only region of concern with step-current changes is that at the trailing magnetic-pole tip. This is the last field a surface point passes through as it passes by the magnetic head. While the ring-head record is primarily longitudinal, a surface region is subjected to an appreciable vertical magnetizing field on both entering and leaving the vicinity of the magnetic-head gap so that some vertical magnetization can be expected. The basic objective in the design of a high-

density magnetic-recording head is to obtain a high reading resolution. This resolution factor is the distance about the gap center line, measured in both directions along the track, over which a recorded step change in magnetization would produce an output signal with surface motion. The guiding concepts in head design for reading stem from a consideration of the gap fringing field. This fact arises from the application of the principle of reciprocity to the recording process. This approach is ideally suited to data recording where the input is composed of step or impulselike current changes. According to the principle of reciprocity, the fringing field at a surface point, established by weakly energizing the *reading coil* (to produce flux densities in the head of the order of those arising on read-back), gives a measure of the sensitivity of the magnetic head to recorded magnetization at that point. Further, the vector components of the fringing field at such a point give the relative sensitivity of the head to the different directions of magnetization. The horizontal component of the fringing field along the surface indicates the head sensitivity to longitudinal magnetization, etc. Thus, for high reading resolution, the fringing field, arising from the above method of energization, must attenuate rapidly on either side of the gap. Then adjacent bits—changes in magnetization—can be placed closer before any mutual interference effects on read-back will occur. On read-back the fields arising from the surface remanent magnetization are very weak, and therefore the head acts, to a close approximation, as a linear element. Since the surface is saturated in one sense or the other and is therefore of low relative permeability, the principle of superposition can be utilized to obtain the net flux linking the head. Each region of the surface in the neighborhood of the gap then contributes to the total head flux an amount proportional to the corresponding fringing field at that point times the intensity of magnetization found there. The net flux would be calculated by integration. The output signal is then obtained by differentiation. By such a procedure it can be shown that for a step change in magnetization, the output-voltage waveform is closely proportional to the distribution of the horizontal component of the fringing-field sensitivity function (for a ring head). A typical read-back waveform resulting from

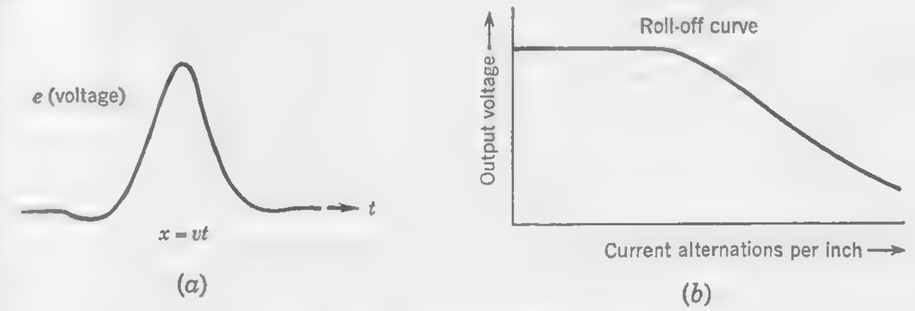


FIG. 12.32. Output-voltage curves.

a steplike reversal in saturation current is shown in Fig. 12.32*a*. Estimates of performance may be made from this waveform by application of the principle of superposition, since reading is predominant in the over-all system characterization. If, for example, the characteristic output signal from a recorded reversal of saturation current has a base width of λ, then the maximum bit density, where no interference from adjacent signals is allowed, would be $1/\lambda$ bpi. On the other hand, if the pulse is symmetrical and the restriction on density is only that no signal-peak variation is to be tolerated, the limiting bit density would be $2/\lambda$.

The above considerations show that the surface magnetic layer, spacing, and the actual magnetic-head design (pole tip shape, coil location, etc.) are very important factors with respect to storage density. This subject will be elaborated upon briefly after noting the fairly direct relation among signal-amplitude reading turns and magnetic-head width. The voltage-output amplitude is directly proportional to the number of turns on the reading coil and the width of the magnetic head. This latter condition arises from the fact that the width of the magnetic head is much greater

than the gap, and hence the fringing flux is oriented almost entirely parallel to the track. Each unit width of the head thus acts itself as a recording member and each contributes proportionally to the total magnetic-head flux linking the encircling reading coil. Since the fringing field about the gap of the magnetic head is not mathematically describable in terms of an elementary functional dependence upon the spatial parameters, no simple statement giving the influence of the variables upon the reading-signal waveform is possible. Normally, the fringing field can be ascertained fairly closely only by recourse to graphical, numerical, or simulation techniques. It is not too difficult, however, to outline qualitatively the influence of the geometrical parameters on the output-voltage waveform. The parameters are defined in Fig. 12.31. The recording layer is magnetized throughout its depth and each differential sheet forming this layer contributes to the total signal output. The fringing field along any given sheet, set up by energizing the read coil (the weighting function), gives a measure of the degree of coupling of the magnetization on this strip with the magnetic head. Then the signal contributions arising from strips located at successively greater spacings from the pole tips of the magnetic head become progressively less in amplitude and broader in width. The latter fact gives rise to conflicting aims in the choice of the thickness of the surface layer. High resolution requires a thin film to avoid a loss in over-all resolution. However, as the storage-layer depth is reduced, there is also a loss in playback-voltage amplitude. For values of d (surface-layer depth) less than the spacing δ, the output voltage is closely proportional to d.

Self-demagnetization, which tends to lower both writing definition and output-signal amplitude, is reduced by a reduction in surface-layer thickness. This result is apparent if the surface, after magnetization, is viewed as being composed of a sequence of horizontal bar magnets. Self-demagnetization is of major significance in a cobalt-nickel surface where a relatively large ratio of B_r/H_c exists. Hence, for equivalent density conditions it is necessary to use a much thinner layer of cobalt nickel than iron oxide. The large value of B_r for cobalt-nickel compensates for the reduction in thickness in an amplitude-output comparison with an iron oxide surface.

In general, an optimum value for d exists. For layers less than this value there is an insignificant improvement in resolution, but a proportional reduction in output voltage amplitude. Nominally for oxide surfaces, the thickness is approximately equal to the spacing in noncontact operation. In contact recording the oxide-layer depth has been fixed by quality, uniformity, and wear considerations rather than by the factors described. A typical range for d for a magnetic-oxide tape is from 0.1 to 0.7 mil. On the other hand, with metallic recording surfaces, the best results are obtained with layer depths of the order of one-tenth those of corresponding oxide layers.

It is obvious that signal amplitude and resolution decrease as the spacing between the magnetic head and the recording medium is increased. This follows directly from a recognition of the attenuation and diffusion of the fringing field with distance from the gap, using the recording-evaluation concepts described earlier for data storage. Hence it is necessary in order to obtain useful storage densities to make δ as small as is mechanically feasible in noncontact operation. Toward this end, magnetic drums are customarily turned (machined) in their own bearings. δ ranges in noncontact recording from a tenth of a mil up to 2 mils. The closer spacings are achieved by the use of air bearings in which the head is supported by the air film between the surface and the magnetic element.

For the idealized magnetic head shown in Fig. 12.33 relations may be written indicating the influence of γ, δ, and d on the output signal produced by a recorded change in saturation magnetization without actually solving for the fringing field. This output is a pulselike signal, and the features to be considered here are the peak amplitude and the pulse width of this signal.

The following series of equations may be written, assuming the magnetization is entirely longitudinal:

$$e(\bar{x}) = d \cdot \bar{H}_x(\bar{x}) \tag{12.5.6}$$

where $\bar{H}_x$ is the average value of H_x (horizontal fringing-field component) through

the surface film. Now where d is not too large a reasonable approximation is to set

$$\bar{H}_x = H(0,0) \tag{12.5.7}$$

Further,

$$H(0,0) = b \cdot \eta \cdot H_g \tag{12.5.8}$$

where $\eta = \dfrac{\delta + d/2}{\gamma}$

H_g = field intensity in gap

b = attenuation of magnetic field as a function of δ, γ, and d (see Fig. 12.33)

b is a monotonically decreasing function and $b(0)$ is somewhat less than 1.0. The parameters δ and γ appear in the manner shown only for this idealized magnetic-head model in which the pole faces and opposing surface are assumed to extend to infinity. Now, if the permeability of the magnetic head is assumed infinite, the total mmf will appear across the gap and hence for the same excitation of the *reading* coil (reciprocity concept) H_g is inversely proportional to γ. Thus $e(0)$ may finally be expressed in the following manner, where d is relatively small:

$$e(0) \propto d \cdot b \cdot \delta/\gamma^2 \tag{12.5.9}$$

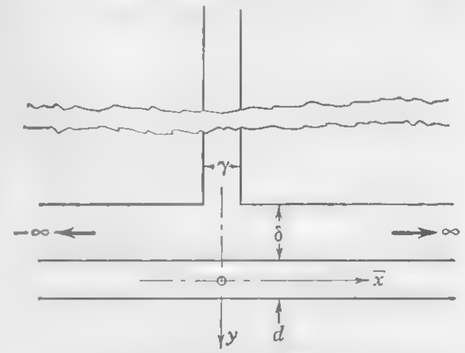


FIG. 12.33. Idealized magnetic ring head.

The peak signal is directly proportional to d only when $\delta \gg d/2$. Then, using this latter condition, if both the gap and spacing are halved, the output signal is doubled. Since in this case δ/γ appears as a ratio which is preserved, this change in δ and γ simply amounts to rescaling the field plot using a factor of one-half. The output-pulse width will be then halved. It is observed that the voltage-time area of the pulse remains constant. This would be expected since this pulse area is equal to the net change in flux linkages, which would be the same. The effects of other changes in the geometrical parameters may be anticipated in a like manner. The fact that it is possible to consider improving the resolution by a factor of two (and hence also the bit density) every time the spacing can be halved has resulted in major effort being placed on improved spacing using air-bearing principles. Nominally bit densities range from 100 to in excess of 1,000 bits per in. as the corresponding hardware uses spacings from 1 mil to 100 μin.

Normally, a component of vertical magnetization might be expected when recording with a ring head. The voltage due to this component is proportional to its magnitude and the vertical fringing-field component. This component of the fringing field is considerably broader than a horizontal one, although it is zero at the gap center. Thus $e(0)$ would not be modified, but a significant degree of vertical magnetization could radically increase the observed pulse width.

So far, these relations concern only spatial considerations. Time-dependent factors are also significant. The frequency bandwidth is established by the recorded information and the velocity of relative motion between the head and surface. In order to place the resonant frequency of the magnetic-head coil beyond the operating-frequency range there is a limitation to the number of turns that can be used. Below the resonant point the head impedance is essentially inductive, and again to decrease the inductive time constant of the head the fewer turns used the better. In the operating range the head impedance will be somewhat linear with frequency. The head is a relatively low impedance device with a d-c resistance of only a few ohms and an inductance of the order of microhenrys in most cases in which the pass band extends to several hundred kilocycles. Typically, the number of turns used is in the range of 100. The required writing current is inversely proportional to N. With recording techniques in which the operating band is shifted away from the low-frequency end, transformers may be usefully employed to match the circuits to the magnetic-head coil(s). Without transformers and with noncontact recording on oxide

surfaces, the write current may be as large as 200 ma. The rapid switching of this fairly large current makes the transient response of the magnetic-head coil quite important since it is not so easy to overdrive and clamp the current to speed up action. In many applications a group of magnetic heads are placed in a stack where very close spacing between them is necessary. This type of use makes interhead shielding necessary to reduce or effectively limit cross talk. Cross talk is the coupling of flux between the heads. A high-permeability magnetic shield (or set of shields) will usually be required in the interhead zones. Eddy-current shielding is also effective. Such shields allow higher track densities for a given signal-to-noise ratio. The effective track width of a magnetic head is almost identical to its physical track width; so the real problem in a multiple-head stack is the mutual magnetic coupling.

Mechanical precision and tolerances are particularly acute with regard to consistent performance. In noncontact recording, the sensitivity of signal amplitude to spacing has been mentioned, and the effects from spacing variation are enhanced as density is increased. On a recorded wavelength basis the function $e^{-2\pi\delta/\lambda}$ gives the relative signal loss as a function of spacing. (Here, λ = wavelength.) This variation in spacing appears as an amplitude-modulation envelope upon the read-back signal from a track. The degree of modulation acceptable sets the runout that can be tolerated. Nominally, magnetic drums with fixed heads are machined to hold the spacing variation within 20 per cent. This means that a magnetic drum storing 100 bpi with a nominal spacing of 1 mil will have a total runout of less than 0.2 mil. Since the signal amplitude is quite sensitive to the spacing δ, the read-back waveform itself provides a sensitive measure of the variation in this parameter. In tape storage, information is recorded parallel by bit (each character) and serial by character. The standard recording tape then may consist of five to seven parallel tracks across the width of the tape. In addition to the spacing control problem, in order that each set of parallel bits may be read in unison, the head gaps must be accurately aligned, and the tape orientation maintained with reference to this line to hold skew to an absolute minimum. Skew refers to the angle between the actual head-gap axis and the normal to a selected tape edge when passing by the head block. The tape is referenced against one edge when traveling by the head array and must ride accurately along this edge. The problem of skew is aggravated when it is necessary that tapes be interchangeable, which is generally the case. On the IBM 727 tape unit, the bit density is set at 200 bpi. At this density and for parallel tracks across ½-in. tape, the alignment between the inside and outside tracks must be held to better than 5 mils offset in 250 mils (¼ in.). The maximum amount of skew between the tape and head assembly is thus less than 0.02 radian. In the Ramac disk array the magnetic element is supported by an air bearing so that the head assembly "follows" the disk runout and holds a spacing of approximately 1 mil. The advantages of such a head support are obvious. In fact, where disk runout may be as great as 50 mils, it would be impossible to consider this type of storage without the use of such a spacing-control mechanism. There are, in this file, only two magnetic heads, one for each side of the selected disk, which are mechanically positioned to the desired track pair. Here, high mechanical accuracy is necessary on track location. This problem is compounded when several accesses are provided for the same disk file. This positioning requirement is alleviated on this machine by providing a wide erase element associated with a narrower read-write magnetic head.

12.5.4. Recording Techniques. Recording techniques broadly refer to the methods of writing and reading binary data. Writing involves the translation of a binary sequence into a unique writing-current pattern and reading involves the detection and conversion of the reading waveform into the corresponding binary information. Data-recording techniques can be divided into two categories, external clocking and self-clocking. External clocking implies a clock track is provided to furnish cell location and writing is synchronized by these timing pulses. These pulses likewise define the sampling interval for each bit period on read-back. Self-clocking, on the other hand, describes recording methods in which clocking information is derived from the recorded data, no external clock track being used.

Subdivisions under these main categories would be the following two basic modes

of storage: (1) techniques that provide a useful output signal for every bit, (2) techniques that provide a useful output signal only for a change in binary sequence or for only one of the two possible values of the binary variable. The first group of codes is in a sense redundant in that the original binary sequence could be reconstructed knowing only the first digit in addition to the changes in digit sequence. This reconstruction requires a timing source closely locked in with the speed of relative motion since the correct number of bits must be inserted in the intervals during which no signal is received from the data track. In situations where the possible variations in binary sequence are limited, this latter group of codes may be applied with a self-clocking mode of operation (using a continuously rephased oscillator).

The output waveform for a given input-current code depends significantly upon the characteristics of the recording unit. It is often possible to use discrete pulses rather than a continuous-current waveform on writing and achieve an equivalent surface-magnetization state due to the fringing of the writing field about the gap.

As mentioned earlier, a knowledge of the basic step, or impulse response of the recording system, allows the read-back waveform to be synthesized for any given binary sequence by employing the principle of superposition. This allows a distinction to be made between the basic resolution merit of the recording unit and the value of a particular recording technique.

The various recording techniques have their own advantages and disadvantages with regard to: the circuits required, sensitivity to sources of noise, reliability, and density considerations. Therefore, the most popular recording techniques will be briefly described and compared with respect to their most salient features.

(*a*) *NRZ Recording.* The conventional *NRZ* (non-return-to-zero) scheme uses a continuous-current waveform when writing with one direction of saturation current throughout a bit period to represent a binary 1, and the reverse sense of saturation current throughout a bit period to represent a binary 0. With this scheme only changes in binary sequence are detected because of the differentiating nature of the reading process. In Fig. 12.34 are shown the input-output waveforms for the binary sequence 110100 at a recording density sufficiently low such that no interference arises between the responses from adjacent current changes. The inherent differentiation process between input current and output voltage is evident. The reading circuit concept illustrated here involves the shaping of the output pulses, the positive pulses setting a flip-flop in the 1 state and the negative pulses resetting this flip-flop. The flip-flop then gates the clock strobes. A gated strobe indicates the presence of a 1 and the absence of such a gated output at clock time denotes a 0.

This type of recording requires the lowest-frequency response for a given bit density and surface velocity. The writing current will almost always have a d-c component. Other observations from an inspection of Fig. 12.34 are the nonsuitability in general to self-clocking and a minimum of crowding between adjacent step-current responses for a given bit density. Normally, an actual read-back waveform will exhibit a base-line ripple due to surface noise as well as amplitude modulation due to a variation in spacing between the magnetic head and surface.

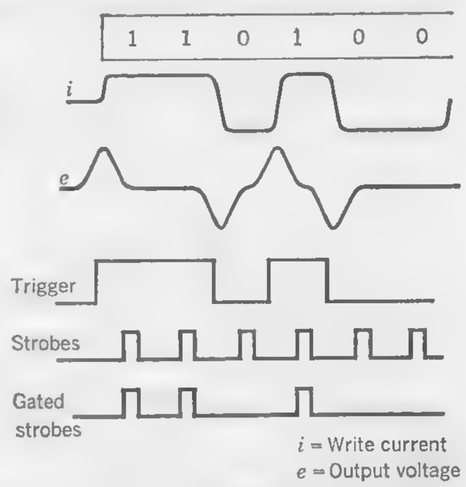


FIG. 12.34. *NRZ* waveforms.

It may be noted that an inherent characteristic of this read-back waveform is the fact that it is alternating in character. This fundamental feature provides the possibility of using logical principles in reading to extend density or increase reliability. With *NRZ* recording if a single pulse is missed which, for example, indicates the beginning of a train of 1's, all the following bits included in the gated block will be

erroneously read. This feature can be avoided by designating each change in current as a given binary digit, e.g., a 1. This coding method is the so-called *NRZI* scheme. Here, if a pulse is missed, only that single bit will be misread.

(*b*) *Pulse Recording.* A second common recording technique is to use a continuous negative saturation current for 0's, and when it is desired to record a 1, a positive saturation current pulse is applied during the selected bit interval. Figure 12.35 indicates a typical current-input waveform and the corresponding output-voltage waveform. It is seen that a double polarity pulse results from a recorded current pulse and, hence, the recording circuits can directly use zero cross-over detection. The voltage response typifies a signal from a recorded "dipole" where the magnetization changes from negative to positive saturation and then back again to negative saturation. A well-known version of the above technique is the so-called *RZ* (return-to-zero) recording in which a 1 is represented by a positive pulse and a 0 by a negative current pulse. In this case a signal is obtained for every bit and therefore lends itself to self-clocking. With pulse recording, where each dipole is to be sensed, every recorded pulse involves two changes in magnetization in contrast to the *NRZ* scheme

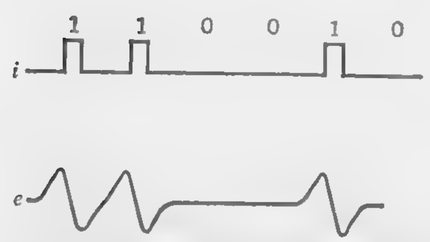


FIG. 12.35. Pulse-recording waveforms.

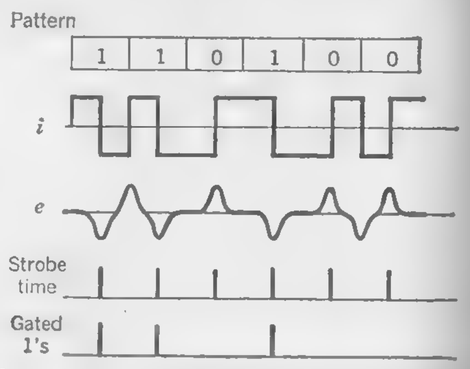


FIG. 12.36. Phase-modulation recording waveforms.

where only a single magnetization change is used to represent a bit of information. Thus, for the same storage density the surface "packing" density is only one-half as great with *NRZ* recording. Pulse-response interference effects occur at lower values of storage density with a *RZ* type of recording, although the possible use of zero cross-over detection is a favorable advantage with this method. One application relying on the pulse-recording method using a negative saturation current for 0's is a single-pass read-write type of operation. The leading pulse of the bipolar pulse response can provide an indication of the state of a cell at a small cyclic time before the information was actually written. This occurs because of the fringing of the writing field forward of the gap. This signal can be used to act logically on this bit of information and based upon this action either rewrite, or not rewrite, a "1" at the time the center of the cell passes by the gap. A short write pulse would, or would not, be applied to the head at this time.

(*c*) *Phase-modulation Recording.* A third recording technique will be referred to here as phase-modulation recording since there exists no universally accepted designation for the method. Whereas in *NRZ* recording the square-wave recording-current bandwidth extends from one-half the bit rate down toward zero, in the phase-modulation techniques the bandwidth extends from one-half the bit rate up to a figure equal to this rate. This latter method then encompasses only one frequency octave while the *NRZ* scheme involves a bandwidth extending over several octaves. This feature of phase-modulation recording is a distinct advantage from a circuit point of view, and further, since no d-c component exists in the writing current, this method is well suited to transformer coupling between the magnetic head and associated circuits. Figure 12.36 illustrates the waveforms obtained from one particular coding scheme using this principle. A satisfactory signal is available for each bit at the low density illustrated, where no overlapping occurs, and hence the method can be used

on a self-clocking basis. At higher densities the half-bit period current envelope may be replaced by a pulse producing essentially the same read-back signal. This arises from the fact that, because of the spread of the fringing field, adjacent pulses of the same polarity will effectively produce a continuous surface magnetization. While the circuits are required to operate at a higher rate here than with *NRZ* recording, this in general will be no great disadvantage. The individual pulse responses on read-back at reasonable densities are more well-defined timewise, as there are only two possible separations between them. The alternating character of the reading waveform has the effect of canceling out the base-line signals of the individual pulses.

(*d*) *Reading.* In practice, many sensing techniques are used, e.g., amplitude, slope, or peak sensing, each characterized by the waveform attribute used. The choice of the particular manipulations to be performed on the read-back waveform is greatly influenced by the over-all system characteristics.

Note the behavior of the roll-off curve (Fig. 12.32*b*) which gives the voltage response as a function of recorded cycles per inch. As a general observation, frequency equalization is applicable as a means of improving the bandwidth character of the "magnetic recording channel" where the original signal-to-noise ratio is more than adequate. Since wavelength response is being compensated by frequency networks, any possible changes in velocity must be anticipated.

Amplitude detection is simple and relatively immune to base-line noise. However, it is very adversely affected by spacing variations due to the recording amplitude sensitivity to spacing, and is limited with respect to high-density recording due to the roll-off characteristic.

By contrast, signal sensing based on a change in waveform slope (peak sensing) will not be sensitive to amplitude fluctuations. Therefore peak sensing is advantageous when large changes in spacing must be tolerated and for high-density recording, since the limitation of the method arises from peak shift affecting timing requirements. Significant peak shift occurs only after considerable pulse crowding. However, this method is more susceptible to base-line noise than is amplitude sensing.

Another reading technique is "energy" sensing in which each pulse amplitude-time area must fall within predetermined limits to be "read." This method gives excellent discrimination against impulse-like noise. There are many other detection techniques of equal or greater merit, depending on the system and its requirements. Here it is only possible to cite the features of inherently different approaches to provide some understanding for comparative evaluations.

In magnetic recording, noise of various sorts is always present, although since distortion in the read-back waveform due to overlapping is in itself highly detrimental, it is common to speak of the desired-signal-to-undesired-signal ratio as the signal-to-noise ratio. Depending upon the noise effects, one or the other of the described detection systems may be advantageous.

12.5.5. Head Selection and Switching. Magnetic drums most generally have a single read-write unit servicing a multiplicity of magnetic heads. To minimize the number of components, switching is usually done directly at the heads. This means low-level switching on reading and relatively high current switching on writing. Also, the switching time between the two operations of writing and reading must be minimized. The reading amplifier requires special precautions to minimize switching and writing transients. These transients will be much larger than the reading signal and could paralyze the reading circuit for an intolerable time. Voltage clamping at the input of the reading amplifier is often used, coupled with a design for minimum recovery time consistent with satisfactory preservation of reading waveform. Crystal diodes are a quite popular means to handle the switching and selection problem. The diodes employed in writing selection must handle in the neighborhood of 100 ma.

In completely parallel operation such as used on magnetic-tape units, these selection features are not present, although switching between the writing and reading mode still exists. Performance requirements (including cost) of a magnetic mass-storage device will dictate the approach to serial versus parallel information flow.

12.5.6. System Descriptions. (*a*) *Magnetic Drum.* As mentioned earlier, a magnetic drum is extremely well suited for service as a working memory for medium-

speed machines. Currently, magnetic drums are predominantly used for this function. While core storage is making some impact, it appears here that high-density drums will find indefinite usage for program storage.

A representative magnetic-drum memory is the unit serving as the internal working store for the California digital computer (Caldic) at the University of California. An aluminum tube slightly greater than 8 in. in diameter and 20 in. in length provides the basic drum. This drum is machined in its own bearings. A 3-mil red oxide magnetic coating (3 M) was put on the drum surface by turning the drum at several hundred rpm and spraying the magnetic paint upon it by repeated passes with a conventional spray gun. The surface was allowed to dry and was lightly buffed between passes. The drum was vertically mounted and directly coupled to a 3,600-rpm single-phase induction motor. This drive gives a revolution time of about 17 msec. Two hundred magnetic heads were mounted in a set of four offset vertical brackets, giving a track density of 10 per inch. A fifth bracket was also located along the drum axis with five heads: one clock track, three spare clock tracks, and an origin or angular reference pulse. All the magnetic heads were set mechanically to a nominal 1.5-mil spacing and then adjusted by signal for equal response through a differential screw adjustment. The bit density is 90 and with a surface speed of closely 1,600 in. per sec, information transfer to and from a memory track occurs at a bit rate of 144 kc. The drum capacity is about one-half million bits with an average access time of 8.5 msec.

Numbers are 10 decimal digits plus sign and are stored in a parallel-by-bit serial-by-digit fashion in the 1-2-4-8 binary-coded decimal scheme. Hence, there are 50 bands of four magnetic heads each. All 50 heads handling a given weighted binary digit (a channel) are serviced by a single writing and reading unit. Switching involves a 1 out of 50 selection. The drum circumference is divided into 2,400 sectors, each corresponding to one word (number or order). The output voltage for the 50-mil-wide magnetic heads is 1 mv per turn and 25 ampere-turns are used for writing. The approximate cost per bit for a magnetic-drum memory such as the one described here is 2 cents. The drum memory is quite rugged and conservative design permits very high reliabilities to be attained.

(*b*) *Magnetic-tape Storage.* Magnetic-tape units are used as auxiliary large-capacity storage units for all high-speed digital computers and are an available option for most medium-speed machines. In the area of input-output equipment they are classified as high-speed units. One special function they can usefully fulfill is to act as a buffer between very-high-speed internal memories and the slow electromechanical input and output devices. Tape interchangeability is an essential systems feature. To update a set of records on a tape it is conventional practice to create a new tape. Large tape libraries and inherent tape replacement are consequent results.

A typical tape system reads information on six to eight parallel tracks on a ½-in.-wide magnetic tape at a bit density of about 200 bits per inch. The tape speed may be 100 in. per sec. In order to handle short lengths of information efficiently the tape can be brought up to speed or stopped in less than 5 msec. The tape may be reversed in less than 10 msec.

The highest-quality magnetic tape consists of a Mylar base (1.5 mils thick) which is coated with a red iron oxide layer 0.5 mil thick. Mylar as a base material is particularly attractive from the point of view of long-term permanent records. The pulleys and capstans are conventional components of the tape-drive system. A moving-coil-operated mechanism starts and stops only a relatively short length of tape near the magnetic-head assembly. Vacuum columns are provided to serve as a tape buffer storage between the drive mechanism and the relatively sluggish tape reels. Uniform tape tension is maintained by storing short tape loops in these columns. The bulk of the tape is thus handled in a fairly leisurely manner. The total tape length may range from 1,400 to 2,400 ft. In the vacuum columns a vacuum is maintained below the short tape loop and atmospheric pressure above it. The columns are divided into three regions and vacuum control switches are used to maintain the tape loops in the center region through control of the tape reel drives. These switches are actuated by the pressure difference maintained across the tape loop.

Skew is the angular measure of the departure of the head-assembly orientation from the normal to tape travel. Skew must be kept below a specified maximum value so that all the seven bits of one character may be read in unison. One edge of the tape is held against a reference plane to maintain this alignment. A full-tape-width erase head is an integral part of the head assembly along with the seven-channel read-write magnetic heads. Mumetal shields are placed between the recording heads to reduce cross talk. A contact pressure of about 6 oz is used between the recording heads and the magnetic tape. Care must be taken to avoid the accumulation of iron oxide particles and dust in and around the region of the recording-head gaps which otherwise would seriously affect tape positioning and head-to-tape spacing. Further, with contact recording in magnetic-tape units, head and tape surface wear will occur. Tape wear can be reduced by avoiding continuous data transfer to and from a limited section of tape. The head-wear problem is handled by a regular replacement schedule. Two techniques which are advantageous in the handling and employment of long lengths of tapes are high-speed block search and tape marking. In the former the information is grouped in blocks with a block-identification symbol preceding each block. Block search is carried out with the head and tape out of contact so that a greatly increased tape velocity is practical. When the desired block is located, contact is reestablished so that the information itself may be stored at the higher densities possible with no spacing. The second technique is to scan the tape initially and insert marks preceding and following all tape defects. These marks will automatically cause the unit to skip over tape defects during operation. By this means, it is not essential that a tape be 100 per cent free of defects in order to be acceptable.

(*c*) *Magnetic Disk File.* A relatively new form of magnetic memory is the magnetic disk file. This unit was designed particularly to meet the requirements of business-data processing. The objectives here are to provide a random-access memory with an extremely large capacity and an access time to any stored record of less than 1 sec. There are disk files with a capacity of 20 million 100-character records with an average access time to any given record of approximately 0.4 sec.

Physically, this disk file consists of 50 disks which are mounted upon a central vertical shaft. Each disk has a 0.7-mil iron oxide recording layer on each disk side. The disks have a 24-in. outside diameter with a recording "band" of approximately 5 in. on each side, giving about a 2:1 ratio between the outer and inner recording-track radii. A 2:1 ratio of outer to inner track radius optimizes disk capacity. A constant information rate is used so that the bit density varies inversely with the track radius, the maximum density (approximately 200 bpi) occurring at the inner track. There are 200 tracks on each disk side, recorded at a track density of 40 tracks per inch. The disks rotate at 1,200 rpm.

The access mechanism drives a pair of magnetic heads, one serving the upper disk sides and the other head the lower disk sides. Somewhat similar to some "jukebox" mechanisms the arm carrying the magnetic heads is first driven vertically along the disk column in the selection of the desired disk, then moved horizontally between the disks to locate the desired pair out of the 200 track pairs available. The given track is then established by head selection. In going to a new address on another disk, the arm is, of course, first retracted from between the disks. Hence the access motion is composed of basically three movements. Mechanical referencing is established by fixed detents. Several such access units can be mounted around the periphery of the disk array, allowing independent references to the file.

The disks invariably have some runout, and for this unit the maximum runout is held to less than 50 mils. It is immediately obvious that mechanical establishment of spacing is impossible and contact operation is prohibited by the high surface speeds involved. Hence, it is of special interest to note that the magnetic heads have their spacing distance from the disk surfaces held to less than a mil by an air bearing. Each magnetic-head capsule is provided with several small nozzles through which a flow of air is provided to produce an air "cushion" between the head and surface. The head is preloaded against the disk and is prevented from coming into contact by the air cushion. For these reasons a small air-compressor unit is an integral part of the file. If air pressure should be lost the head unit is designed to snap back automatically.

In order that the head-arm unit may enter between a pair of disks, the magnetic heads are not brought against the disk surfaces until track location has been accomplished. The use of a film of air to maintain a specified noncontact separation between a magnetic head and a recording surface greatly extends the number of feasible mechanical structures for magnetic memories. An even more promising development in this area is the use of a so-called "gliding shoe" to support the magnetic head and hold spacing. The principle here is that of the Kingsbury bearing, relying on the boundary layer of air carried along by a moving surface. The head forms a wedge with the recording surface with the air film between them holding the spacing against a positively applied load on the shoe directed toward the surface. The successful operation of a gliding shoe is critically dependent upon the mounting of the shoe and the surface finish and contour of the opposing surfaces.

The magnetic unit on the Ramac consists of a wide-erase element followed by a narrow read-write element in order to relax the requirements for precision upon track location. The erase unit is energized only at writing time in order to eradicate all previously recorded information. All information to and from the disk file is transmitted through a core buffer for synchronization purposes.

(*d*) *Future Trends.* Other magnetic devices are being worked upon to capitalize upon the advantages of magnetic recording for data storage. Among recently announced equipment, bit-density specifications range up to 1,500 bits per in. The important role played by magnetic recording in the development of electronic data-processing systems now appears to be only a beginning in the expanding future of this form of mass storage.

12.6. WILLIAMS-TUBE MEMORY SYSTEM*

12.6.1. Basic Arrangement. In this type of memory, digits are represented by charge distributions which exist on small areas of a cathode-ray-tube screen. The digit areas, or bits, are usually arranged in a 32- by 32-bit square to give a storage capacity of 1,024 bits per tube. A "signal" or "pickup" plate, made of metal sheet or gauze, is attached externally to the face of the tube as shown schematically in Fig. 12.37.

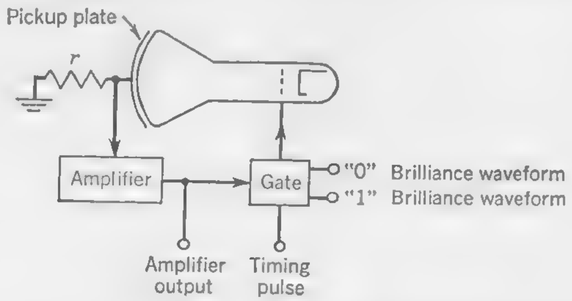


FIG. 12.37. Regenerative storage system.

Thus each digit area is capacitance-coupled into a common channel. A suitable amplifier enables signal voltages, developed across a resistance r between the pickup plate and ground, to be observed.

The amplified signals are fed to a gate circuit which, under the control of basic timing waveforms, delivers to the cathode-ray-tube grid waveforms designed to write either a "1" or a "0."

12.6.2. Technological Theory. The secondary emission ratio of the screen material used in commercial cathode-ray tubes is greater than unity if the tube is operated in the normal range of, say, 1,000 to 3,000 volts. It follows that if, operating under such conditions, the electron beam is switched on to a single spot on the cathode-ray-tube screen, the number of secondary electrons leaving the spot and moving toward the internal conductive coating will exceed the number of primary electrons arriving

* By Frederic C. Williams.

at the spot. The net loss of negative charge causes the potential of the bombarded spot to become positive with respect to the remainder of the screen, and there is then a force exerted on all emitted electrons tending to direct them back to the spot. An equilibrium is established with the spot about 2.5 volts positive with respect to the remainder of the screen, such that the net secondary current is equal to the primary current, i.e., the effective secondary emission ratio is unity. Since the capacitance of the spot is very small, this equilibrium is reached in less than 1 μsec.

A second equilibrium of negligible importance to the practical operation of the storage system occurs over a period of many seconds after the initial switching on of beam current, as low-velocity secondaries, causing virtually no secondary emission, strike the screen surface remote from the spot, causing the whole screen to stabilize at about −15 volt with respect to the internal conductive coating.

The above discussion is summarized in Fig. 12.38*a*, the important feature being that the area of screen under the spot shown shaded at (1) is positive with respect to the mean screen potential. Increasing positive potential is plotted in the direction of the arrow, so that, using the analogy of gravitational field, electrons may be said to fall toward regions of positive potential. This potential distribution is called a "well."

If two spots shown at (1) and (2) in Fig. 12.38*b* are bombarded alternately two independent wells are formed with little or no mutual interference provided that the centers of the spots are separated by more than about 1.5 spot diameters. If, however, the spot separation is less than this critical value, as shown at *c*, then considerable mutual interference occurs as follows: If the spot is initially in position 1, a well of normal type is "dug" in this position, but on moving to position 2 some of the electrons emitted during the digging of a well there are attracted by the positive potential of well 1. Thus well 1 is partially refilled, as indicated by the continuous line in position 1 of Fig. 12.38*c* (i). When spot 1 is again bombarded the reverse situation occurs, a full well being left in position 1 and a partially filled well in position 2, as shown at *c* (ii). This process of digging one well and partially filling the other can be repeated indefinitely, and if the system is symmetrical the charge ejected from one well is equal in magnitude to that deposited in the other, since the charge ejected was deposited during the previous half cycle of operation. Indeed, if the digging of one well and refilling of the other had the same time scale, this process would produce no signal at the pickup plate since there would be no net rate of change of charge on the screen surface.

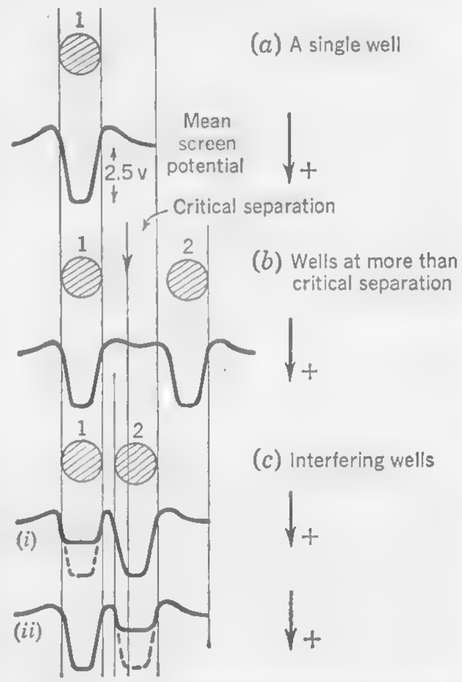


FIG. 12.38. Potential distributions—the "well."

However, the refilling of a well, which may take about 4 μsec, is much slower than the digging of a well since less than half of the secondary electrons emitted have a component of velocity in the direction of the well being refilled, and many of these have velocities too great to be attracted to that well or any part of the screen. Thus signals are received at the pickup plate, and these will now be considered.

In Fig. 12.39 the grid-modulating signal is shown at *a* and the shift waveform necessary to produce the two spots 1 and 2 at *b*. Digging a well establishes additional positive charge on the screen surface and causes conventional current to flow to ground through the resistance r of Fig. 12.37, producing the positive pulses shown at *c* in Fig. 12.39. Filling a well reduces the positive charge on the screen surface, and a negative pulse is obtained as shown at *d*. The areas under the waveforms at *c* and *d*

are the same, because the charges involved are equal, but *d* has a longer time scale and smaller amplitude.

A further effect, not yet considered, is caused each time the beam current in the tube is changed. When the beam is switched on a cloud of electrons in the secondary current, and in the beam itself, is introduced in the vicinity of the pickup plate, and a transient current must therefore flow to the plate to supply the required induced positive charge. A negative pulse is thus obtained at the pickup plate, to be followed by a similar positive pulse when the beam current in the tube is switched off and the electron cloud is dispersed. These pulses, whose shape and time scale are ultimately defined by the transient response of the amplifier, are shown at *e* in Fig. 12.39.

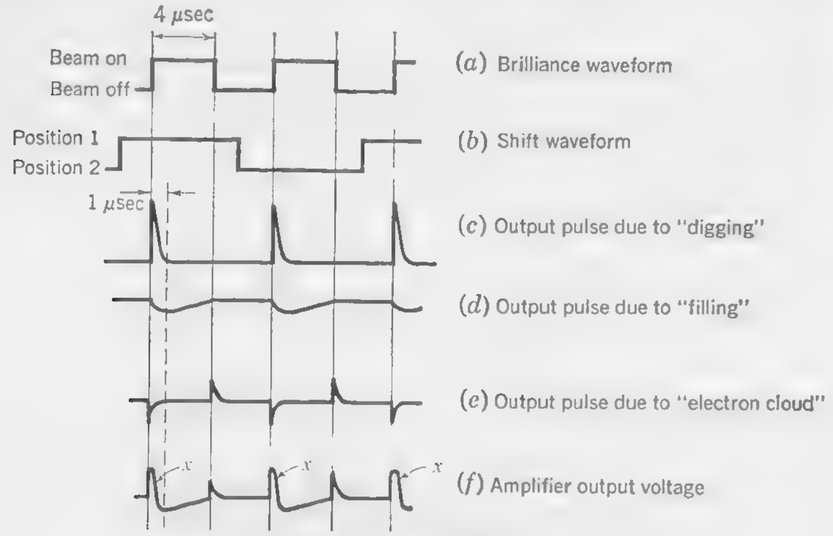


FIG. 12.39. Double-dot signals.

When, therefore, the partially filled well is bombarded the net output voltage is the sum of the waveforms *c* due to digging the well, *d* due to filling the adjacent well, and *e* due to the electron cloud, giving the waveform shown at *f*. The important feature of this waveform is the net positive pulse *x* immediately following the instant at which the beam is switched on.

This positive pulse contrasts with the negative pulse obtained at this instant if an isolated well is rebombarded, since the waveforms *c* and *d* do not exist in this case, and the output is solely due to the electron-cloud effect *e*.

From the phenomena described the following statements can be made:

1. Either of two states of charge may be left at will at a given spot on the cathode-ray-tube screen. These states are
   - *a*. A well of full depth, by bombarding the storage spot, and not subsequently bombarding any other spot within interfering distance (see Fig. 12.38*a*)
   - *b*. A partially filled well, by bombarding first a storage spot, and then another spot within interfering distance [Fig. 12.38*c*(i)]
2. Renewed bombardment of the storage spot will give at the instant of recommencing bombardment a negative signal from the pickup plate in case 1*a* (see Fig. 12.39*e*), or a positive signal in case 1*b* (see Fig. 12.39*f* at *x*).

Item 1 above indicated the mechanism of writing a digit and item 2 the mechanism of reading it. It has been assumed throughout that no leakage of charge has occurred between successive bombardments of a storage spot, though charge distributions will be maintained for only a short time—a few tenths of a second—depending upon surface leakage. In practice this is overcome by arranging for each stored digit to be read and rewritten well within the inherent storage time, thus giving a new start to the stored charge, with all leakage compensated—a process called "regeneration."

Many systems of operation based on these phenomena are possible, the ones which will now be described being selected because they are the most extensively used. The systems differ fundamentally only in the displays used to represent "0" and "1," the regeneration process being common to all.

12.6.3. Practical Systems. (*a*) *System 1—Double-dot Display.* With the shift waveform shown in Fig. 12.40*a* either a dot or a double dot can be displayed on the cathode-ray-tube screen as at Fig. 12.40*f* and *h* using the brilliance or write input waveforms *b* and *c*, respectively. Corresponding with *f* the negative pulse *g* occurs immediately after time $t = 0$, whereas corresponding with *h* the positive pulse *i* occurs as explained in the preceding section. These pulses are selected from the amplifier waveforms by the gate *e* and are used to provide the read output.

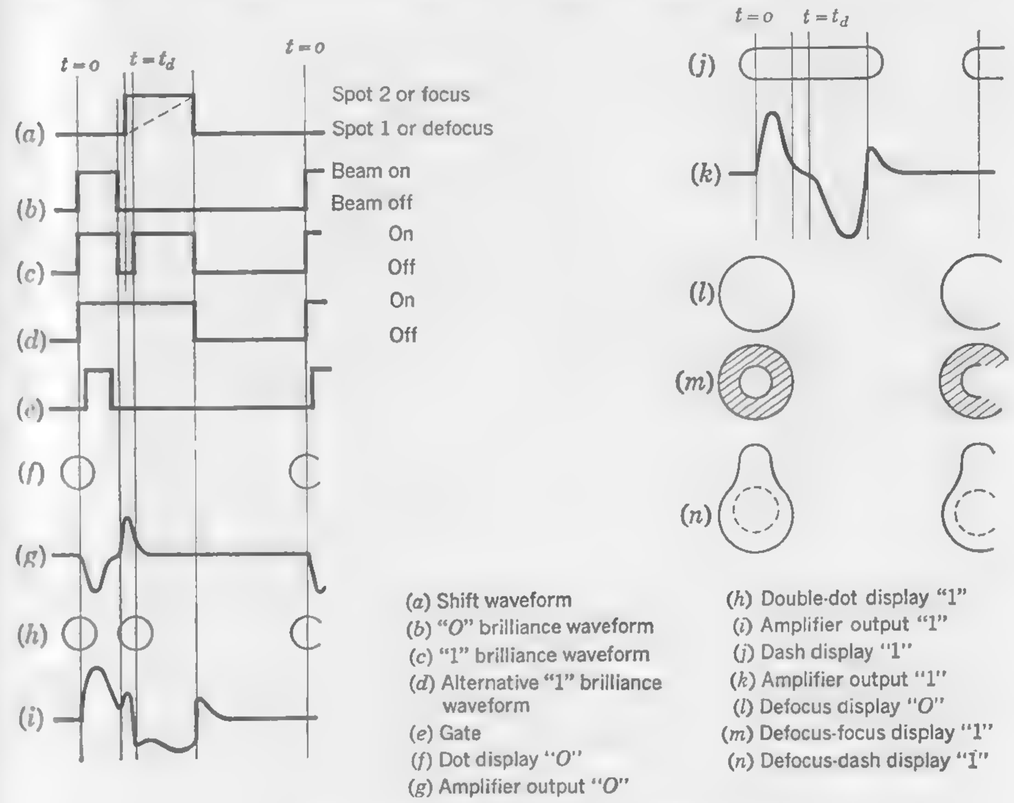


FIG. 12.40. Alternative practical storage systems.

To make the system regenerative, it is necessary to cause dots or 0's to be rewritten whenever dots are read, and double dots or 1's to be rewritten whenever double dots are read. That this procedure is possible may be seen from Fig. 12.40, because reading is complete during the gate time, but the decision to write "0" or "1" need not be taken until $t = t_d$, since the "0" and "1" brilliance waveforms are identical until this time. Hence, if the cathode-ray-tube brilliance waveform is fed through a gate circuit which is controlled by the gated amplifier output in such a way that beam current is turned on again at t_d (see Fig. 12.40*c*) if the control signal is positive, but not (see Fig. 12.40*b*) if the control signal is negative, the system will be regenerative in that it will immediately rewrite everything it reads. This is shown schematically in Fig. 12.37.

The double-dot system was used in the 40-digit parallel stores of the Institute for Advanced Study computer and other computers derived from it.

(*b*) *System 2—Dot-dash Display.* In the double-dot system the crucial positive pulse in the amplifier output corresponding with a 1 is the sum of three pulses. If,

in particular, the negative contribution due to filling the adjacent well (see Fig. 12.39*d*) is eliminated, then a correspondingly greater positive-signal output is obtained.

This is achieved by replacing the double dot by the dash display shown in Fig. 12.40*j*, which is produced by the ramp function shown dotted at *a*. The representation of 0 remains a single dot, but 1 is represented by a dash, filling of the initial storage well taking place as the spot moves along the dash. A fully dug well is left at the end of the dash, but this is made to be outside the critical distance, and substantially no filling of this well occurs when the initial storage well is rebombarded.

The amplifier output signals are shown at *g* and *k* on Fig. 12.40.

(*c*) *System 3—Defocus-focus Display.* In this system 0 is represented by a defocused spot, and 1 by a defocused spot followed by a focused spot in the same position as shown in Fig. 12.40*l* and *m*, in this case the waveform *a* being applied to the focus electrode. Thus a large well is dug initially, and the annulus shown shaded at *m* is filled only if a 1 is required.

The negative-signal amplitude *g* is the same as in the previous systems, and the positive-signal amplitude *k*, which is proportional to the area of the annulus, will be the same as that for the dot-dash system if the ratio of the radii of the defocused and focused spots is $\sqrt{2}$:1.

The main advantage of this system is that filling of the annulus occurs from all directions round the focused spot, whereas in the previous systems unidirectional filling occurred—from right to left in Fig. 12.40*h* and *j*. Hence the method offers greater protection against screen blemishes. These impurities or pinholes, which acquire large negative potentials, act as barriers to the refilling process and thus reduce the positive-signal amplitude. In a typical comparison of systems, a screen imperfection which was sufficient to reduce to zero the useful positive-signal amplitude in the dot-dash system only reduced the corresponding signal amplitude in the defocus-focus system by 10 per cent.

A secondary advantage of this system and those to be described is that less precision is required of the deflection circuits in returning to a given spot since initially the spot size is greater than the focused spot of the initial dot in the previous systems.

(*d*) *System 4—Circle-dot Display.* The main disadvantage of the defocus-focus system is that there is a tendency at storage areas near the edge of the raster for the focused spot to be displaced from the center of the defocused spot by deflection defocusing, so that filling of the remoter parts of the defocused well presents some difficulty. Two methods of removing this defect are in common use.

In the first a circle is drawn, using a focused spot, by imposing high-frequency sinusoids 90 deg apart on the vertical and horizontal deflection plates. This circle replaces the annulus in Fig. 12.40*m*. The annular well can be filled or not, as desired, by a focused spot so that reinspection with a subsequent circle will produce either a positive or negative pulse as before.

This system is used in the BESK computer in Sweden in a 40-digit parallel store with 256 digits per tube.

(*e*) *System 5—Defocus-dash Display.* The second method of removing the effect of the off-center focused spot is to make the spot contract from the defocused to the focused state relatively slowly by applying the ramp function shown dotted at *a* in Fig. 12.40 to the focus electrode. If, during the contraction, the spot is also moved, say vertically, the defocus-dash display shown at *n* is obtained to represent 1, 0 still being represented by a defocused spot. For a 1, the defocused well is now filled over its entire area, and a small (focused) well is left outside the confines of the original well. With suitable dimensions, this small well can be left within interfering distance of the large well, and in the case where a 0 must be written over a previously stored 1 it assists the operation by allowing a negative going "fill" pulse to occur during the subsequent regeneration of the 0.

12.6.4. Operating Factors. (*a*) *Speed.* Though laboratory tests have obtained storage, for example, with the double-dot system at 500 kc per sec, the present practical operating speed of all the systems is about 100 kc per sec. Typical figures for the widths of the dig and fill pulses of Fig. 12.40*c* are 1 and 4 μsec for the dot-double dot system, and 1.8 and 4 μsec for the defocus-dash system. Since a further interval

of about 2 μsec must be allowed for the shift waveforms to move the spot to a new digit position, substantial improvement on a digit period of 10 μsec is difficult.

(*b*) *Read-around Ratio.* Having decided the speed of operation, the number of digits which can be stored per tube is considered. In assessing this, it is not possible to use the distance of critical separation, namely, 1.5 spot diameters, which applies to two wells bombarded alternately. In a complete raster of digits, a well representing a stored 0 can be filled, not only by adjacent bombardment within its own digit area but also by bombardment during reading, writing, or regeneration of adjacent digit areas. For the usual mode of operation, alternate digit periods are assigned to regeneration and action (reading or writing). The stored digits are visited sequentially during regeneration periods and as instructed during the interleaved action periods. In particular, the same digit area may be selected during each action period; so that this is bombarded n times during action periods between regenerations of the adjacent stored 0 in the raster where n is the total number of digits stored per tube. In the worst case, that of most bombardment, it is necessary to store a single 0, every other digit in the raster being a 1, including the action digit which is adjacent to the 0. If N is the number of times that it is permissible to bombard this action digit without destroying 0 then n cannot be greater than N for reliable operation. N has been called the "read-around" ratio, and this effect provides a severe limitation on the number of digits stored per cathode-ray tube.

It should, however, be noted, first, that it is possible to relax the condition $n \not> N$ in a parallel store by restricting the number of references to a particular digit area either by machine design or by suitable programming, and secondly, that in a serial store the necessary value of N is very much reduced. In the latter case, if N_s is the number of times that it is permissible to bombard an action line of digits, without destroying an adjacent line of stored 0's, all other digits in the raster being 1's, then for a square raster $\sqrt{n}$ cannot be greater than N_s. Thus in the United States satisfactory parallel storage of 1,024 digits per tube is being obtained using selected commercial tubes, for example, the 5CPIA tube with the double-dot system, and in England serial storage of a similar number of digits was obtained using the 6-in. CV1097 tube with the dot-dash system.

Some improvement in the read-around ratio is obtained by adopting suitable arrangements of the digits on the tube face. To avoid large variations in spot size due to deflection defocusing, the digits are sometimes arranged in the form of a cross. It is more usual, however, to use a square array, hexagonal packing being obtained by staggering alternate rows by half of the separation between digits. In the double-dot and dot-dash system a further improvement in digit density is obtained by using alternate directions of displacement for the second dot or the dash.

A method of improving the read-around ratio by integration of the amplifier-output signals can be applied to all systems, and it will now be described.

(*c*) *Improvement of Read-around Ratio by Integration.* The partial filling of a well corresponding with a stored 0, during bombardment of adjacent area or areas, causes the negative amplifier output shown by the full line in Fig. 12.41*b* to become slightly positive as indicated by the dotted line. It is this effect which determines the read-around ratio by making it eventually impossible to distinguish between the dotted line at *b* and the positive pulse at *e*. Although the initial negative pulse at *b* is still present, it is impossible to find a satisfactory strobing period because the peak of the initial positive pulse *e* occurs at the later period as shown by the vertical dotted line. However, if the signal at *b* is integrated during the dig pulse, it is the net area of the signal during this period which is important, and as shown by the dotted line at *c* this may still be negative. The corresponding waveform for a stored 1 is shown at *f*, and the position of the strobe, which is now easily chosen, is shown at *d*.

The read-around ratio is now determined by the much less severe criterion that it must be possible to discriminate between the net areas of waveforms *b* and *c* during the period of the dig pulse.

(*d*) *The Amplifier.* The integration required above may be achieved by making the value of *r* in Fig. 12.37 equal to 500 kilohms, and with the inevitable input capacitance of about 50 $\mu\mu$f a time constant of 25 μsec results. During the dig pulse of 1 to

2 μsec, the effective input impedance to the amplifier is almost purely capacitive, and the input voltage is the integral of the signal current flowing from the pickup plate. In a particular case the amplifier is a simple four-stage voltage amplifier with a nominal bandwidth of 5 to 750 kc per sec, and an input signal of 0.8 mv (peak amplitude of the positive pulse) gives an output signal of 40 volts. Incorporated in the fourth stage is a shorted delay line equal to a half of the dig pulse period. By this means the benefits of integration are preserved during the dig pulse, but any long-term components caused by integrating are greatly attenuated, since the delay line causes the difference between the input waveform and that waveform delayed by the dig-pulse period to be produced. The output of the fourth stage is fed via a cathode follower to a clamp circuit, which is clamped at all times except during the strobe period.

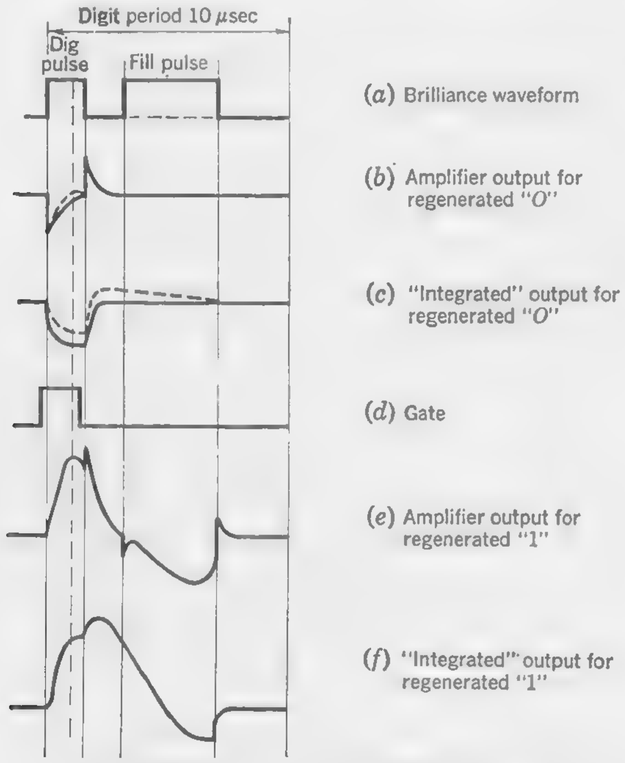

Digit period 10 μsec
Dig pulse
Fill pulse
(a) Brilliance waveform
(b) Amplifier output for regenerated "0"
(c) "Integrated" output for regenerated "0"
(d) Gate
(e) Amplifier output for regenerated "1"
(f) "Integrated" output for regenerated "1"

FIG. 12.41. Waveform of recommended storage system.

Integration of the amplifier-output signals makes the best use of the available information, but in the final analysis the accuracy of the information depends on the cathode-ray tube and particularly on its screen surface.

(*e*) *Special Cathode-ray Tubes.* Intensive cathode-ray-tube development has taken place with the primary objective of eliminating screen imperfections. A secondary objective has been to decrease the spot size to obtain the resultant increase in information density. Clearly these two aims are to some extent incompatible, since a given screen imperfection has a larger effect with a smaller spot, and a compromise must be made.

In the United States this activity has produced the 3-in. IBM 79 or 3VPI tube operated at 2,500 volts, the spot size being 12 mils. Two out of three tubes made have no screen imperfections, and this tube is being used with the dot-dash system. In England, a similar development differing only in emphasis has produced the 6-in. VCRX266. For this tube, the operating voltage is intentionally restricted to 1 kv, to reduce the effect of screen imperfections, and the spot diameter is 15 mils. Screen imperfections are permitted within stated safe limits, and their effects are removed by the use of systems 3, 4, or 5.

Each tube can operate in the parallel mode with a read-around ratio of 512, which, applied to computing machines, means that 1,024 digits may be stored per tube without placing any restriction on the program.

12.6.5. Conclusion. It is difficult to decide finally on a best system of digit representation; the five systems described have all been used. The systems most extensively used are the dot-dash system in the United States and the defocus-dash system in Europe.

Improvements such as integration, digit arrangement, and packing, and the development of special tubes have been exploited in increasing the permissible read-around ratio and in obtaining greater tolerance and reliability in service.

12.7. MAGNETIC-CORE STORAGE AND SWITCHING TECHNIQUES*

Magnetic-core storage units driven by transistors are used in most computers to mechanize the random-access storage function. Initially these units were tube-driven, physically large, and costly and required special air conditioning to maintain a constant temperature around the cores. They had the supreme advantage of high reliability, compared with all other available forms of random-access storage. Designers and users suffered the expense and inconvenience of the early units in order to obtain the high reliability. Core-storage units of recent design are generally smaller and less expensive than the early units. Greater reliability has been achieved in the newer core memories. Typical tube-driven core-storage units with a capacity of about 200,000 bits were sold in production quantities at $1 to $2 per bit. A present-day unit of this capacity can cost less than 25 cents per bit. The volume of an early unit might be over 300 cu ft, not including the air-conditioning unit and motor generator. Recently manufactured units do not need either of the latter devices, and they range in volume from less than 60 cu ft down to as little as 1 cu ft. Power consumption has dropped by at least a factor of 10.

Early units were held at a temperature controlled within ±10°F. Most of the present units operate from 0 to 40 or 50°C, and some operate from −20 to 100°C. The reliability of a typical early core-storage unit is such that a mean error-free time of up to 1,000 hr can be obtained with 4 per cent maintenance. With present-day units, mean error-free times in excess of 1,000 hr can be obtained with only ½ per cent maintenance.

These changes in magnetic-core storage have taken place over a period of 4 years, and the art is still progressing rapidly. Core-storage units with capacities of less than one million bits are now more reliable and, in most cases, more economical than any other form of electronic storage.

During this 4-year period, the importance of different aspects of the core-storage art has shifted. Magnetic switches were originally of great importance but are now of minor value in most random-access storage units. Recently a demand has developed for small special-purpose storage units, frequently classified as buffers. A major problem encountered in designing buffers is the need to reduce the transistor count to a minimum in order to achieve high economy and reliability. Special types of magnetic switch have been used to solve this problem. The magnetic switch has in fact almost completely superseded the transistor as a logical timing or decoding element in small storage units. Moreover, techniques have been developed whereby utilization of magnetic switches to perform logical and timing operations has been extended beyond the storage unit into the computer itself. This is not true magnetic-core logic but is rather the introduction of logical properties into memory units.

The evolutionary process illustrated by the use of magnetic switches in buffers is continuously taking place, and new applications for obsolete techniques are being found every day. Therefore, all aspects of core storage, whether or not they are considered important at present, will be discussed in this section. The number of different core-storage systems is too great to be fully covered in a short description. Many new storage devices are being investigated, and development of existing devices

* By Raymond Stuart-Williams. This material is reproduced by courtesy of Ampex Corporation, Redwood City, California, who reserves all United States and foreign rights with respect thereto.

and systems is continuing. For these reasons, fundamental concepts and principles will be expounded to a greater extent than is usual in a handbook. Actual data presented will be restricted to widely used techniques and components.

Storage Cores

12.7.1. Characteristics of Storage Materials. The ideal storage material would have the characteristics shown in Fig. 12.42. Its hysteresis loop would be absolutely rectangular and its saturation characteristics completely flat. A major hysteresis loop and a series of minor loops of a good commercially available material are shown in Fig. 12.43. These loops, particularly the minor ones, are not too far from the desired rectangular form.

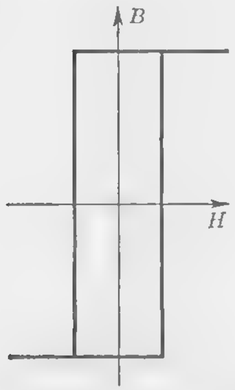


FIG. 12.42. Hysteresis loop of an ideal storage material.

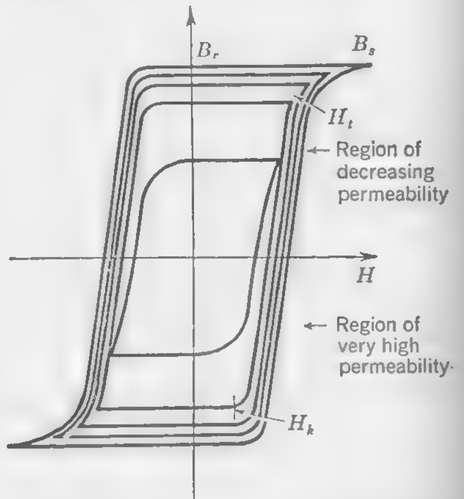


FIG. 12.43. Major and minor loops of a practical storage material.

The minor loops are almost invariably employed in coincident-current storage applications. If the core is in one state of saturation and H is altered to tend to move the core to the other state of saturation, at first there is very little change in flux. A value H_k is finally achieved beyond which the permeability is high, and this region continues to a value H_t. When H is reduced to 0, the remanent flux density is equal and opposite to the original flux density. If H_k/H_t is greater than 0.5, the core is a useful storage element since there will be almost a trigger action in moving from one flux-storage state to another. There is a range of values of H over which adequate storage loops are obtained. In a good storage material this range is high, allowing the designer to tolerate relatively large changes in driving current in the storage system. If the core is driven by means of a rectangular pulse which generates the value of H_t corresponding to the center of this range, the core will take a time T_s to switch from one state of storage to the other. This time is known as the characteristic switching time, or turnover time, of the material.

12.7.2. Switching Times of Storage Materials. In any given storage material the switching time is related to the value of the coercivity (H_c) of the loop under consideration by the equation

$$T_s(H_c - H_o) = K \tag{12.7.1}$$

where H_o and K are constants depending on the material. This equation does not hold when H_c becomes very nearly equal to H_o. It has been found experimentally that for mid-range values of H_c and T_s most storage materials fall near a curve represented by

$$T_s(H_c - 0.3) = 1 \tag{12.7.2}$$

where T_s is in microseconds and H_c is in oersteds. This equation can be used to

predict roughly the performance of as yet unknown materials, provided that T_s is less than 5 to 10 μsec.

Equation (12.7.2) seems to present an ultimate limit to the rapidity of storage operations using conventional materials and methods. Recent work on special ferrites and thin metallic films has shown that this empirical equation is incorrect and that it is possible to make materials which will operate at very high speeds without requiring extremely large drive currents.

In the case of thin metallic films, the film is made to operate more rapidly than is indicated by the theoretical equation by employing an internal magnetic stress in the material to assist in obtaining rapid switching action. Under these circumstances switching is achieved in a dipole manner instead of in the normal domain-wall-movement manner which is usually associated with ferromagnetic phenomena. A price is paid for this increase in speed in that the flux change that occurs during switching is reduced relatively rapidly as the speed is increased. At very high speeds, virtually no flux change occurs and the thin metallic film does not really switch in the normal ferromagnetic sense. Instead a slight angular change in the vector of magnetization occurs, which is exhibited externally in a manner similar to a flux change.

Special ferrites which operate more rapidly than the ferrites used in coincident-current storage applications attain a high speed both by having special properties and by being used in a special mode of operation. A typical BH loop used in this mode is shown in Fig. 12.44. The broken loop is a relatively minor loop of the material. The partial loop shown in full, which is employed for storage applications, does not exhibit normal rectangular-loop properties. Storage points are indicated by the numbers 0 and 1. To the left of the B axis the material is sufficiently rectangular so that coincident-current techniques can be used to determine whether a 0 or 1 is stored. However, to the right of the B axis no rectangular properties are exhibited and a special mode of operation is required. In this mode, a large pulse is used to destroy all information in the selected cores and, at the same time, read the contents of these cores. (A complete description of this mode of operation will be given later.) It is possible to design special materials for this type of operation which will operate approximately ten times as fast for the same drive as would a normal coincident-current storage material.

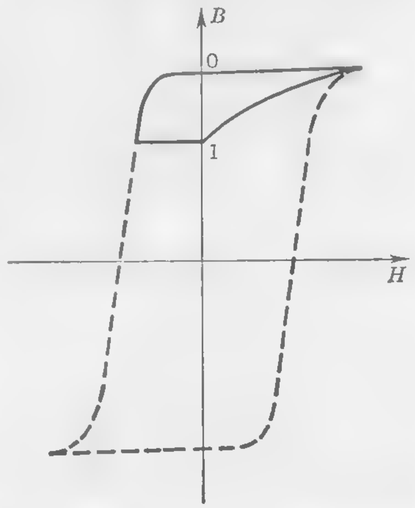


FIG. 12.44. BH loop for partial-flux mode of operation.

12.7.3. Types of Storage Materials. *Thick Metal Tapes.* Typical materials in this category are Deltamax and Orthonol. They have a very high saturation flux density, high Curie point, and low coercive force. They have extremely rectangular characteristics, which are lost when the metals are rolled into thin films. Metal tapes of this type begin to be unusable in storage and switching applications when their thickness is below 0.001 in. They are useful in applications where slow operation is desired and where relatively high power must be transmitted. At high rates of operation, eddy-current losses become very large.

Thin Metal Tapes. Typical materials in this category are 4-79 molybdenum-permalloy and HI-MU-80. These materials have half the saturation flux density of the thick metal tapes, a high Curie point, and low coercive force. They are not particularly rectangular in tapes thicker than 0.001 in. Their rectangularity and coercive force increase as the material is rolled into thin tapes. The 0.00025- and 0.000125-in. tapes have excellent properties. Cores wound with thin tapes have been used in special storage systems, but their chief applications are as switches, shift registers, and core logic. The characteristic switching time depends on tape thick-

ness; in the very thin tapes this time can be as low as a few microseconds. It is difficult to maintain high uniformity in rolling thin tapes. Components must therefore be designed using the maker's specifications and tolerances and not by means of measurements or tests on a small sample batch.

Thin Films. Thin metallic films are usually deposited by electrolytic or vacuum-evaporation techniques, whereas thin tapes are almost always rolled from a block of the metal. Although thin films have extremely desirable qualities, they present considerable difficulty in controlling the uniformity of the separate storage elements. The present investigations into thin films are aimed at finding a thin metallic element which will exhibit relatively little dynamic loss, and at establishing techniques to improve the electrical properties by utilizing magnetic or mechanical stress in the material. To solve this latter problem, four major approaches are being investigated. The first is to use very thin, flat disks of material deposited in a magnetic field so that they exhibit a built-in magnetic stress. These disks are deposited, usually in vacuum, a large number at a time on a glass plate. Printed windings laid over the top of the disks provide a means of access to them. The "twistor" provides another method of approaching the problem. In this approach, the material is plated or deposited on a wire and is then subjected to a mechanical torsional strain in order to set up a helical field pattern. Access to this type of storage device is by means of windings wrapped around the wire and by the use of the wire itself. A third approach is to deposit the material on small rods so that it is formed into a bar magnet, the effective thickness of which is very small compared with its length. And, finally, an attempt is being made to construct thin metallic cores, which are analogs of the conventional ferrite core, by depositing a thin layer of metallic material as a continuous path around a small toroid. At present, insufficient equipment has been built utilizing any of the four techniques described above to be able to predict which, if any, of them will achieve wide usage.

Ferrites (Ferrospinels). Cores of these materials are made by several manufacturers under different trade names. Most of them are magnesium-manganese or copper-manganese ferrite bodies. In manufacture, the oxides or other salts of the metals are mixed in the correct quantities and then undergo a series of grinding, milling, and firing cycles. The product at this stage is a uniform free-flowing powder. Cores are pressed individually from this powder under considerable pressure and then subjected to a lengthy firing process under controlled atmospheric conditions and at a maximum temperature which may be in excess of 2500°F. The cores, ceramic in nature, are vitrified throughout and consequently are very hard and brittle. They are reasonably good insulators and exhibit high permittivity but are relatively weak magnetically. The maximum flux density is low, as is the Curie point. The coercive force is usually rather high. Since cores are insulators, they may be molded in fairly large sections without altering their loss characteristics. In a tape core a large amount of the cross-section area is taken up with intertape insulation and a bobbin. In a ferrite core the entire cross-section area is useful material; this compensates, to some extent, for the low flux density.

Magnesium-manganese ferrites are exceedingly stable. Extensive tests indicate no change in their properties with life. If correctly packaged, they do not appear to be subject to subsequent mechanical or electrical failure. The experience with well-packaged ferrite cores indicates that the probability of failure is less than 1 part in 10^{10} and probably is 1 part in 10^{12} core hours. The copper-manganese ferrites, though fairly stable, have the undesirable characteristic that at high temperatures their properties can change. In all cores of this type that have been tested, a temperature of 100°C appears high enough for a slow change to commence. However, in all normal equipments the copper-manganese ferrites can be considered highly reliable. Most commercial ferrite cores will operate reliably up to about 70°C. Special cores which will operate in a coincident-current mode at temperatures in excess of 100°C are also available commercially. Experimental cores have been produced which can be operated at temperatures higher than 200°C.

All ferrite cores have relatively large temperature coefficients, which are predictable, stable, and uniform. Each parameter of the core's operating characteristics has a different temperature coefficient. The parameter of major importance is the drive current required to maintain the operating characteristics constant over a range of temperature. In most magnesium-manganese ferrite cores this temperature coefficient is approximately ½ per cent per degree centigrade. The drive current must be reduced as temperature is increased by this amount, or conversely must be increased by this amount as temperature is reduced. It is possible either to temperature-compensate the whole system so that drive currents are optimum at the operation temperature, or alternatively to operate the cores at all times at the highest expected temperature.

An important property of a ferrite core is the effective tolerance available over a range of drive current and temperature. This tolerance is usually expressed as the percentage ratio of H_k to H_t. For example, a core is said to have a 65 per cent knee. This implies that a tolerance of 15 per cent is allowable on the half current fed to the core in the coincident-current mode without expecting deterioration of the operating properties of the core. A typical core which operates over a wide temperature range will have a knee of 75 per cent at 25°C and will still be in excess of 65 per cent at 70°C. This indicates that the core could be operated without temperature compensation over a wide range of temperatures. However, adequate temperature compensation of the system will tend to center the core at the optimum operating current, allowing the driving elements to deteriorate considerably with age without affecting performance of the equipment.

Because ferrites can be molded by mass-production methods into small-sized cores with a high degree of uniformity, they are almost always adopted in storage applications. They also find some use in switch cores and in shift-register and logical applications. Ferrites are not suitable for large switch cores; since they are insulators, it would be difficult to remove heat from the cores. Cores have been made in a range of materials having characteristic switching times from 0.1 to 15 μsec, but commercially available materials have switching times from 0.2 to 5 μsec.

Table 12.2 lists the properties of two typical ferrite materials which are in common use today.

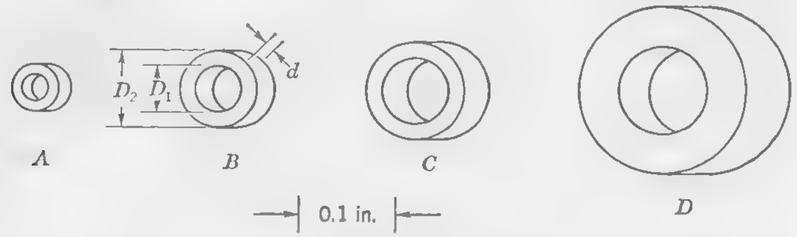
Table 12.2. Properties of Two Typical Ferrite Materials

	1-μsec material	4-μsec material
Switching time, μsec	1	4
Coercive force, oersted, H_c	1.25	0.55
H_k (approx)	1.125	0.51
H_t (approx)	1.54	0.92
Saturated flux density	1,740	2,100
Remanent flux density B_r, gauss	1,600	2,000
B_r/B_s	0.92	0.95
Curie temperature, °F	450	360
Temperature coefficient, % of H_c per °F	0.14	0.17–0.2

12.7.4. Core Geometry. It is well known that in a long, open-ended permanent bar magnet some portion of the flux tends to return down the magnet. This flux has a demagnetizing effect on the magnet which tends to reduce the rectangular properties of the material. In thin films where, almost universally, open-ended storage elements are employed, the ratio of thickness of the magnet to length is always extremely small. In ferrite or metallic toroids, this ratio is usually relatively large. It is not practicable to cut and rejoin either metal or ferrite cores; for the resulting gap, however small, will reduce the B_r/B_s ratio and increase the coercive force. Since even small cracks or chips will alter the properties of cores, the toroids must be complete and undamaged.

Several typical toroids are shown in Fig. 12.45. The current required to drive the toroid is a function of the mean diameter; consequently, the tendency is to use the

smallest possible core. Practical limitations determine the minimum wire size for a given toroid. The number of wires which can be inserted through the center of the toroid will be proportional to $(D_1)^2$ if the core is threaded at right angles to its plane. In practice, the plane of a core is normally at 45 deg to the direction of the wire threading; therefore, the actual effective aperture is considerably smaller than the maximum available aperture shown in Fig. 12.45. Aperture size is also affected by



Core	D_2	D_1	d
A	0.050	0.030	0.015
B	0.080	0.050	0.025
C	0.100	0.070	0.030
D	0.180	0.090	0.045

FIG. 12.45. Typical toroids.

the thickness of the core. For this reason, cores should be as thin as is feasible in order to permit easy passage of wire through the aperture. The ampere-turns required to drive the toroid will be proportional to $D_1 + D_2$. The voltage drop per turn will be proportional to $d(D_1 - D_2)$. If minimum current to drive the toroid is desired, a large diameter should be used. As the diameter of a toroid is increased, the number of turns which can be wound through the window rises more rapidly than the number of ampere-turns which are required to drive the toroid. For a fixed wire size and given geometry, the driving current is inversely proportional to the diameter of the toroid. The voltage drop increases rapidly as the number of turns is increased; therefore, multiple-turn windings are employed only in small storage units. Most switches are treated as small matrices employing large cores and multiple-turn windings.

The core is not a perfect thin-walled toroid. As the value of H is reduced in moving from D_1 to D_2, a new storage loop is adopted which is narrower and has less flux change than the original loop. Consequently, the over-all storage loop of the core is the sum of a number of loops which have different values of coercive force and of flux density. Consider a toroid consisting of three coaxial rings. Let the BH loops of these rings be L_1, L_2, L_3, as shown in Fig. 12.46*a*. The over-all BH loop of the system will be that of Fig. 12.46*b*. If an infinite number of rings, each of which is

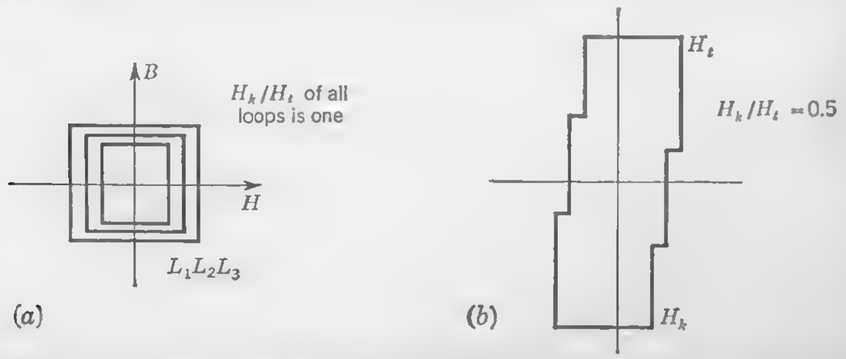


FIG. 12.46. Effect of wall thickness on toroid characteristics. (*a*) Loops of three coaxial rings. (*b*) Over-all system loop.

perfectly rectangular, is taken, the over-all loop will be a parallelogram. The rectangular characteristics of the core are degraded as the ratio of D_1/D_2 is reduced. Manufacturing considerations limit the wall thickness. A D_1/D_2 ratio of 0.7 is about the maximum value which will give adequate yield. For a high yield of the most economical cores, $D_1 - D_2$ should not be reduced below 0.03 in. Fortunately most ferrite materials used in storage applications have very good rectangular characteristics. This makes it possible to use relatively inferior core geometry and still attain good operation. However, if as nearly ideal as possible characteristics are required, a thin-walled toroid should be chosen.

Ferrite cores which have been adopted as relatively standard cores are listed in Table 12.3.

Table 12.3. Standard Cores

D_2	D_1	d	D_1/D_2	$\frac{1}{2}(D_2 + D_1)$	No. of wires wound through window
0.100	0.070	0.030	0.700	0.085	5–15
0.080	0.050	0.025	0.625	0.065	4–11
0.060	0.038	0.018	0.633	0.049	4–8
0.050	0.030	0.015	0.600	0.040	4–6
0.030	0.020	0.006	0.666	0.025	3–5

12.7.5. Typical Ferrite Cores. Manufacturers of ferrite cores specify core material and geometry in order that components of definable electrical and mechanical properties may be supplied. A number of typical commercially available cores are described below. Considerably more data about these cores can be obtained from the manufacturer.*

Core 1. The size of this core is 0.080 in. (outside diameter), 0.050 in. (inside diameter), 0.025 in. (thickness). Its switching time is less than 1.25 μsec. The drive pulse required is normally 0.82 ampere-turns. At 25°C the core is usually driven with this amount of full current by means of a pulse which is 1.5 μsec long and has a rise time of 0.2 μsec and a fall time of 0.3 μsec. The switching time P_s is 1.25 μsec. The amplitude of a 1 signal (dV_1) is 125 mv; and the amplitude of a 0 (dV_s) is 10 mv. Peaking time t_p is 0.65 μsec. The signal-noise ratio at peaking time is at least 50:1. This indicates that the 0 or noise signals decay rapidly as the amplitude of the 1 signal builds up. At peaking time the amplitude of noise or 0 signal is less than 2.5 mv.

Table 12.4. Test Conditions

Temperature	25 ± 1°C
Drive pulse I_R or I_W	740 ma ± 1%
Drive pulse I_{PR} or I_{PW}	470 ma ± 1%
Pulse rate	56 kc ± 1%
Pulse duration	9 μsec
Pulse rise time (10% . . . 90%)	0.20 ± 0.02 μsec
Pulse overshoot	1% max rise or droop

Table 12.5. Output Signals (under Test Conditions)

Disturbed 1 dV_1	90 mv min
Peaking time t_p of dV_1	0.65 ± 0.05 μsec
Switching time t_s of dV_1	1.40 μsec max
Peak disturbed 0 dV_s	25 mv max
Amplitude of dV_s at t_p	0.5 mv max

Typical test conditions and output signals obtained under these conditions are shown in Tables 12.4 and 12.5.

* Ampex Computer Products Co.

Sample cores are also tested over a range of temperature to determine temperature-drive relationship. Results are summarized in Table 12.6.

Table 12.6. Temperature-drive Relationship

Temperature	0°C	25°C	40°C
$I_R = I_W$, ma	900	820	775
$I_{PR} = I_{PW}$, ma	550	510	460
dV_1 min, mv	125	125	125
dV_z max, mv	30	30	30

Core 2. The size of this core is 0.080, 0.050, 0.025. It is intended for use in low-speed memories where economy of current drive is important. The core normally switches in less than 4 μsec when driven with a full current of 364 ma. At 25°C the drive pulse is usually 6 μsec long with about 1 μsec rise and fall time. Under these conditions the switching time is 3.8 μsec. dV_1 is 35 mv, dVz is 4.5 mv, peaking time is 2 μsec, and the signal-noise ratio is in excess of 100. Typical test conditions and the output signals obtained under these conditions are shown in Tables 12.7 and 12.8.

Table 12.7. Test Conditions

Temperature	25 ± 1°C
Drive pulse I_R or I_W	330 ma ± 1%
Drive pulse I_{PR} or I_{PW}	200 ma ± 1%
Pulse rate	56 kc ± 1%
Pulse duration	9 μsec
Pulse rise time (10% . . . 90%)	0.6 ± 0.05 μsec
Pulse overshoot	1% max rise or droop

Table 12.8. Output Signals (under Test Conditions)

Disturbed 1 dV_1	20 mv min
Peaking time t_p of dV_1	2.0 ± 0.2 μsec
Switching time t_s of dV_1	4.0 μsec max
Peak disturbed 0 dV_z	7.0 mv max
Amplitude of dV_z at t_p	0.5 mv max

Results of tests performed over a range of temperature to determine temperature drive relationship are summarized in Table 12.9.

Table 12.9. Temperature-drive Relationship

Temperature	0°C	25°C	40°C
$I_R = I_W$, ma	400	360	345
$I_{PR} = I_{PW}$, ma	240	215	200
dV_1 min, mv	30	30	30
dV_z max, mv	7	7	7

Core 3. This core is one of the small-sized (0.050, 0.030, 0.015) variety. Switching time is 1 μsec when pulsed with a full current of 500 ma. At 25°C the core is driven with a pulse of 500 milliampere-turns full current and a width of 1.5 μsec. It switches in less than 1 μsec and peaks at about 0.5 μsec. dV_1 is 75 mv, dV_z is 5 mv, and signal-noise ratio is in excess of 100.

Typical test conditions and the output signals obtained under these conditions are shown in Tables 12.10 and 12.11.

Table 12.10. Test Conditions

Temperature	25 ± 1°C
Drive pulse I_R or I_W	450 ma ± 1%
Drive pulse I_{PR} or I_{PW}	285 ma ± 1%
Pulse rate	56 kc ± 1%
Pulse duration	5 μsec min
Pulse rise time (10% . . . 90%)	0.2 ± 0.05 μsec
Pulse overshoot	1% max rise or droop

Table 12.11. Output Signals (under Test Conditions)

Disturbed 1 dV_1	45 mv min
Peaking time t_p of dV_1	0.55 ± 0.05 μsec
Switching time t_s of dV_1	1.25 μsec max
Peak disturbed 0 dV_z	10 mv max
Amplitude of dV_z at t_p	0.5 mv max

A typical temperature-drive relationship over a range of temperature appears in Table 12.12.

Table 12.12. Temperature-drive Relationship

Temperature	0°C	25°C	40°C
$I_R = I_W$, ma	565	500	480
$I_{PR} = I_{PW}$, ma	345	310	285
dV_1 min, mv	65	65	65
dV_z max, mv	8	8	8

Figure 12.47 shows the relationship of output and drive for core 3. Similar curves can be plotted for most ferrite cores. It will be seen that, at low values of drive current, the operation of the core is limited by a low value of the 1 signal. At high values

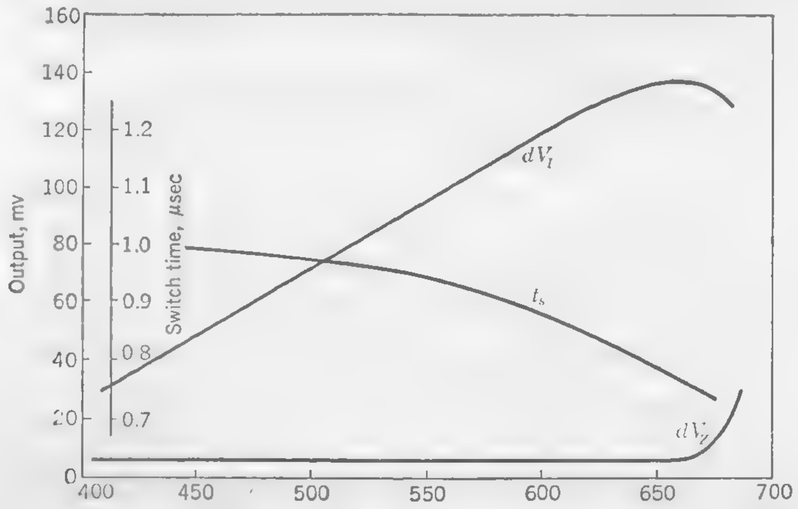


Fig. 12.47. Output-drive curves for typical ferrite core.

of drive current, the disturb or 0 signal dV_z rises rapidly while the 1 signal dV_1 drops rapidly, and there is a marked deterioration in the operation of the core.

Core 4. The dimensions of this core are 0.050, 0.030, 0.015. Its switching time is nearly 3 μsec when pulsed with a full current of 234 ma. In normal operation it is

pulsed with either the full current or a half current of 117 ma, using a pulse with slow rise and fall times and a width of 3 μsec. This core is particularly useful in transistor-driven applications and does not require the use of costly transistors. dV_1 is 15 mv, dV_z is 2.5 mv, peaking time is approximately 1.5 μsec, and the signal-noise ratio is in excess of 100. Test conditions and output signals are shown in Tables 12.13 and 12.14.

Table 12.13. Test Conditions

Temperature	25 ± 1°C
Drive pulse I_R or I_W	212 ma ± 1%
Drive pulse I_{PR} or I_{PW}	138 ma ± 1%
Pulse rate	56 kc ± 1%
Pulse duration	6 μsec min
Pulse rise time (10% . . . 90%)	0.6 ± 0.05 μsec
Pulse overshoot	1% max rise or droop

Table 12.14. Output Signals (under Test Conditions)

Disturbed 1 dV_1	10 mv min
Peaking time t_p of dV_1	1.6 ± 0.05 μsec
Switching time t_s of dV_1	3.2 μsec max
Peak disturbed 0 dV_z	3.5 mv max
Amplitude of dV_z at t_p	0.5 mv max

A typical temperature-drive relationship for a range of temperatures is shown in Table 12.15.

Table 12.15. Temperature-drive Relationship

Temperature	0°C	25°C	40°C
$I_R = I_W$, ma	225	235	217
$I_{PR} = I_{PW}$, ma	155	148	132
dV_1 min, mv	14	14	14
dV_z max, mv	4	4	4

Core 5. This 50-mil core (0.050, 0.030, 0.015) is intended for applications where high environmental temperatures are encountered. The characteristics of this core are shown in Tables 12.16 through 12.20.

Table 12.16. Recommended Drive Conditions

	At 25°C	At 70°C
Drive pulse, ma-turns	430/215	326/163
Pulse width, μsec	2	2
Pulse rise time, μsec	0.2	0.2
Pulse fall time, μsec	0.3 min	0.3

Table 12.17. Typical Output Signals

	At 25°C	At 70°C
Switching time t_s, μsec	1.0	1.1
Amplitude 1 dV_1, mv	40	40
Amplitude 0 dV_z, mv	10	8
Peaking time t_p, μsec	0.5	0.5
Signal-noise ratio at t_p	100	100

Table 12.18. Test Conditions

Temperature	25 ± 1°C
Drive pulse I_R or I_W	420 ma ± 1%
Drive pulse I_{PR} or I_{PW}	264 ma ±1%
Pulse rate	10 kc ±1%
Pulse duration	5 μsec
Pulse rise time (10% . . . 90%)	0.2 ± 0.05 μsec
Pulse overshoot	1% max rise or droop

Table 12.19. Output Signals (under Test Conditions)

Disturbed 1 dV_1	30 mv min
Peaking time t_p of dV_1	0.55 ± 0.05 μsec
Switching time t_s of dV_1	1.2 μsec max
Peak disturbed 0 dV_z	10.5 mv max
Amplitude of dV_z at t_p	0.5 mv max

Table 12.20. Temperature-drive Relationship

Temperature	25°C	55°C	80°C
$I_R = I_W$, ma	435	345	296
$I_{PR} = I_{PW}$, ma	266	211	180
dV_1 min, mv	35	35	35
dV_z max, mv	10.5	10.5	10.5

Core 6. This is a 50-mil core (dimensions 0.050, 0.030, 0.015) designed for use in word-selection or linear-selection memory systems with high repetition rates. The rates are achieved by utilizing a small portion of the total flux, thereby minimizing heat-dissipation requirements. This core is pulsed with a large current I_R in order to read and with a long low-amplitude pulse I_W in order to write. If a 1 is to be written, a digit drive pulse I_D is applied in addition to I_W. Recommended operating conditions at 25°C and typical output signals are shown in Tables 12.21 and 12.22.

Table 12.21. Recommended Operating Conditions at 25°C

Current pulses	Read I_R	Write I_W	*Digit* I_D
Amplitude, ma-turns	500	170	130
Rise time t_r, μsec	0.1	0.1	0.1
Fall time t_f, μsec	0.1	0.1	0.1
Width at base t_d, μsec	0.3	0.6	0.6

Table 12.22. Typical Output Signals

	Amplitude, mv	Peaking time t_p, μsec	Switching time t_s, μsec
Read 1 (dV_1)	100	0.1	0.25
Read 0 (dV_z)	20		

12.7.6. Testing Storage and Switch Cores. Every core that is used in a storage unit must be tested thoroughly to ensure as near perfect operation as is possible in the matrices and switches. Ferrite cores are subjected to three series of tests: production sampling, core evaluation, and quality-control tests.

Sampling tests are performed on a sample of cores in a production run to determine statistically whether all the characteristics of the batch are correct. The entire production run is then tested for critical operating parameters such as loop squareness, switching time, peaking time, and operating-current range. Storage cores are tested on an automatic testing device which, in view of the large number of cores involved, is faster and more economical than hand testing. A standard core is used to maintain calibration of the tester, which is usually set to operate at a fixed ambient temperature. Each core is demagnetized before testing commences to remove any previous "history."

Of all the evaluation tests for storage cores, the most important is the determination of the operating-current range. A core is tested for drive range by means of a long train of variable-amplitude pulses and with various sensing amplifiers and automatic grading circuits. The core is first disturbed with a low value of current to determine whether it provides sufficient voltage output. It is then disturbed by a high value of current either as a long pulse or as a train of pulses in the direction which could cause demagnetization. Conditions of low turnover current and high disturbed current are those most likely to demagnetize the core. The amount of demagnetization compared with the signal produced on complete turnover is known as the "disturb sensitivity" of the core. For most storage systems, the high and low ends of the operating range are defined as ±5 to ±15 per cent of the optimum value, and the allowable disturb sensitivity is 0.3 or 0.4. Testing for disturbed signals is of particular value when cores with very large 1:0 ratios are desired.

Switch cores, both ferrite and metal, are ordinarily tested by hand because only small quantities are used. A core is energized by current pulses which will switch it in a time somewhat shorter than the characteristic switching time. The area of the output pulse is examined to measure the flux output, and the shape of the pulse is examined to determine the coercive force and the behavior at the knee of the loop and in the region of decreasing permeability. The over-all loop is also checked with a *BH* tester. When the switch core is wound as a separate transformer, it is tested in an analog of the circuit in which it will operate. Testing of switch cores is generally complex because the principal parameters are indicated by the shape of the output pulses. Skilled human operators are more effective than machines in performing the necessary tests.

The final series of tests for both storage and switch cores are the quality-control checks. Storage cores are tested automatically on the testing device and switch cores are tested semiautomatically. A double check of operating parameters is usually made on a sample of cores at this point to eliminate any defective batches which may have been accepted as the result of a malfunction of the automatic equipment.

Automatic testing equipment is expensive and, to justify its cost, must be capable of testing a group of cores over long intervals with substantially the same results. All the circuits in the equipment must be extremely precise because even a slight change in pulse shape can alter test results.

Storage Techniques

12.7.7. Principles of Coincident-current Selection. A magnetic core is a nonlinear element which will not change its state when a half current $I/2$ passes through it but will change its state when a full current I, consisting of two half currents, is applied. This property of the core is the basis for the principle of coincident-current selection of a core from an array for storage or read-out of information. It is possible to select a specific core by energizing the horizontal X wire and the vertical Y wire through the core, each wire carrying a half current. The cores on the X and Y coordinates which are linked by only a half current will not change state, but the core at the intersection of the wires will receive the full current and will change its state. In practice, cores will tolerate more variation in current than the theoretical 2:1 ratio because the knee of a core is usually at 65 to 75 per cent of the I value. Currents which are not exactly equal may therefore be used in the X and Y wires, particularly for the purpose of improving the tolerance of the electronic part of the system.

Selection by triple coincidence is theoretically possible by using X, Y, and Z wires through the cores. A current of $I/3$ would be applied to each of the selected coordinate wires, and only the core at the intersection of the energized wires would change its state.

A simple coincident-current matrix is shown in Fig. 12.48. Conventionally, if a current corresponding to $H_t/2$ is applied to two of the intersecting wires, the core at the intersection will be forced into the 0 state. Similarly if $-H_t/2$ is applied, the opposite state of storage will be produced. If, for example, B and D are energized with a positive half current, then core 3 will be forced into the 0 state. If the core is already in the 0 state before this action occurs, very little flux change will occur; but if the core is in the 1 state, a complete flux reversal will occur. If all the cores are linked by a sense winding such as EE, the flux change resulting from a core being forced from the 1 state to the 0 state would induce a voltage in EE. This voltage could be detected by means of a suitable amplifier connected to the terminals EE. This is the basic method of reading employed in a coincident-current memory. In order to write, positive half currents can be applied to the desired lines to set the

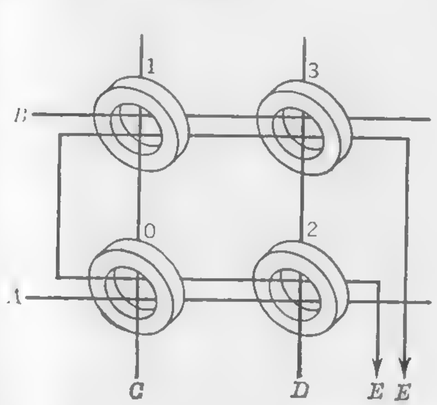


Fig. 12.48. Core matrix for simple coincident-current operation.

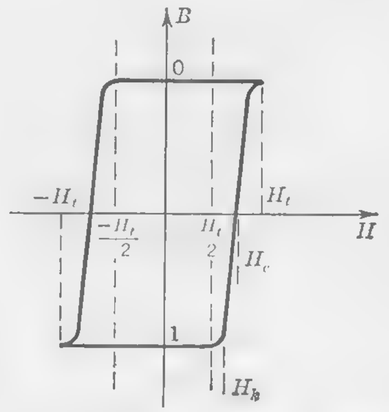


Fig. 12.49. Hysteresis loop of a core operated by coincident current.

selected core into the 0 state, or negative half currents can be applied to set the core into the 1 state. In most memories a writing action consists of first setting the core into the 0 state and, if it is desired to write a 1, later setting the core into the 1 state.

Coincident-current selection is illustrated in Fig. 12.49 in terms of a hysteresis loop. The half currents correspond to $H_t/2$ or $-H_t/2$. Full currents correspond to either H_t or $-H_t$. In order to obtain the best tolerance in the system, ideally the most suitable core to employ is the one in which H_k exceeds $H_t/2$ by as large a factor as possible. It should be noted that H_k and H_t are special terms employed in coincident-current storage. The normal parameter associated with magnetic materials, the coercive force, is shown as H_c. Usually the values of H_c and H_k are very close, while H_t is much larger than H_c. For this reason, H_c and H_k are frequently confused in the literature and are often used indiscriminately to replace each other.

In a coincident-current matrix, a sense winding links all the cores. Undesired signals introduced by all the half-excited cores induce a voltage in the sense winding. During a reading action in an n^2 matrix, one core will receive a full excitation and $2(n-1)$ cores will receive a half excitation. The other cores receive no excitation. If the cores had perfectly flat saturation characteristics, the half-excited cores would provide no output. In practice the voltage induced in the sense winding by a half-excited core may be an appreciable fraction of the voltage induced by the selected core, the amount of voltage depending on the type of core and on the type of operation. The output due to half excitation takes the form of a voltage pulse which lasts for a short period of time. The peak amplitude of this pulse may be as great as a third of the amplitude of the turnover signal. If the sense winding linked all cores

in the matrix in the same polarity, a considerable voltage would be induced in this winding because of the disturbed cores. In a 10,000-core matrix, for example, the total voltage induced in the sense winding would be sixty times as large as the turnover or 1 signal. In order to overcome this problem, the sense winding is arranged to link the matrix cores in alternating polarities so that equal numbers of cores provide positive and negative disturbed signals with a consequent tendency toward cancellation. The disturbed signals can be made to cancel completely if the matrix is well designed, the cores are uniform, and the correct driving functions are performed.

Large-capacity storage units sometimes employ magnetic-core switches or transformers to drive the matrices, which are usually operated in a parallel manner. Core switches and transformers must be restored after they have been set, and two pulse times must be allowed for any transformer or switch operation. The cycle which is employed under these circumstances consists of two parts. First, a reading operation is performed. If it is part of a reading cycle, the information is read from the selected core; or if the operation is part of a writing cycle, cores are turned over to the 0 state. The switch or transformer cores are usually set during this read operation. Secondly, a write operation is performed during which the cores are usually set into the 1 state. The transformer or switch cores are usually reset during this write operation. Writing is used to generate information destroyed during reading or to insert new information into the storage device. An additional driver known as the digit-plane or inhibit driver is energized if a core is to remain in the 0 state during the write part of the cycle. This driver provides a half current to an inhibit winding in the storage matrix which prevents the 1 setting action inherent in a normal write operation.

The same cycle is used for both reading and writing operations. In reading, the second beat of the cycle is used to restore the information which was read. In a parallel-storage system, a number of matrices corresponding to the number of bits in the stored word are driven in cascade. Separate digit-plane drivers, provided for each matrix, are used during the second beat of the cycle to modify the information in the individual bits of the word. In some storage units, the two beats of the cycle must always occur; they are usually referred to as the "reading" and "writing" beats. In other units it is possible to separate the two beats. They are then referred to as the "unload" and "load" operations. If this latter mode of operation is used, an unload operation must be preceded by a load operation, and similarly a load operation must be preceded by an unload operation. The basic principles of operation of a core-storage unit do not require that an unload operation must occur first and then immediately be followed by a load operation. In many devices considerable advantages may be obtained by allowing random access not only to the address in the unit but also to the part of the cycle that is required.

Triple Coincidence. Coincident-current operation has the advantage that the number of wires required to select a core is less than the number of cores which can be selected. In normal operation $2n$ wires will select n^2 cores. If a material is used in which the knee is greater than 67 per cent, then $3n$ wires would select n^3 cores. Triple coincidence may be achieved by a biasing technique without using a material that is any more rectangular than those normally used in double coincidence. Triple coincidence is not of great value in small matrices, but as the size increases it may become necessary to adopt triple coincidence.

There is a continuous demand for ever higher rates and ever larger capacity in storage units. In transistor-driven equipment, high rates can be achieved only by direct connection of the transistors to the storage arrays. The tendency is to increase greatly the number of transistor drivers as rate and capacity are increased. Under these circumstances, triple rather than double coincidence is the economical choice, considering the number of drivers required (see Table 12.23).

Materials are available which can be used in simple triple-coincidence applications. The signal-noise ratio achieved with these materials is poor, and there is an undesirable loss of tolerance in operating the system. Because of these disadvantages, triple-coincidence systems are rarely used. When such systems are considered essential for storage appplications, the bias technique mentioned above is almost invariably adopted.

Table 12.23. Number of Drivers for Double- and Triple-coincidence Matrices

No. of cores in matrix	No. of drivers	
	Double coincidence	Triple coincidence
64	16	12
4,096	128	48
262,144	1,024	192
1,000,000	2,000	300

12.7.8. Operation of a Number of Matrices in Parallel. A form of three-dimensional operation can be achieved by operating a number of matrices in parallel. One set of drivers drives all the matrices in cascade, and an individual driver per matrix sets the information into that matrix. This system is not so efficient as a true triple-coincidence system, but it does have the advantage that a whole character or word can be operated on at once with a consequent increase in computing rates. Even in storage units which work in conjunction with serial machines, it is common practice to employ a parallel-storage unit together with circuits which perform serial-parallel conversion. For example, a 262,144-bit store could employ one matrix using 48 drivers or sixty-four 4,096-bit matrices using a total of 88 drivers. The second arrangement would be 64 times as fast as the first, although the increase in the number of drivers is less than a factor of 2.

Two basic methods have been adopted to control the insertion of information into the individual matrices in a parallel system: use of a digit-plane or inhibit winding, and use of a switch.

Control by a Digit-plane or Inhibit Winding. In this method, the outputs of the magnetic or transistor switches are taken through all the matrices in cascade. Both switches are operated at the same time and provide like outputs. The selected core in every parallel matrix is set to 0 and the reading operation is performed. At the end of this operation the two selected switch cores are in the nonnormal state of flux. These cores are then restored to their normal state, which tends to set all the selected matrix cores into the 1 state. A fourth winding is wound through each matrix, linking all the cores in the same sense. An mmf equal and opposite to that flowing in each of the selection lines can be applied by this fourth winding during the period when the switch cores are restored. When this mmf is applied, the selected core receives a net half excitation and remains in the 0 state. The fourth winding, known as a digit-plane or inhibit winding, is driven by a digit-plane driver. This makes it possible to control the individual bits of the word independently, even though the major selection and drive process is performed by one set of drivers. This action is illustrated in Fig. 12.50. In this diagram and in subsequent diagrams, a pulse is written subscript p if the mmf is tending to move the core into the 0 state and subscript n if the mmf is tending to move the core into the 1 state. It should be noted that the introduction of the inhibit-winding pulse causes considerable mmf change in all the cores linked with this pulse. They first move in a positive direction, then move negative, and finally move positive again. Because this action introduces

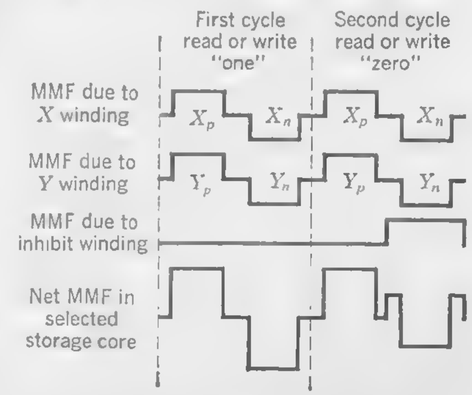


FIG. 12.50. Cycle when inhibit winding is used to operate matrices in parallel.

considerable noise and ringing in the magnetic array, a pause is required after a digit-plane operation before the next reading operation can be started.

Control by Switch. In this method, there is one X switch common to all the parallel matrices and a separate Y switch for each matrix. One set of drivers links all the Y switches in cascade. The X and Y drivers, operating simultaneously, set all the selected Y switch cores. The selected matrix cores are set to 0, and reading is performed. A restoration winding linking all the cores in each of the Y switches is driven by a digit-plane driver so that the Y switches can be reset independently. In order to insert a 1 into a particular matrix, the Y switch controlling that matrix is restored at the same time as the X switch. In order to insert a 0, the Y switch is restored at some time other than during the X-switch restoration. This action is illustrated in Fig. 12.51. In most storage units in which this type of operation is employed, it is the normal practice to restore

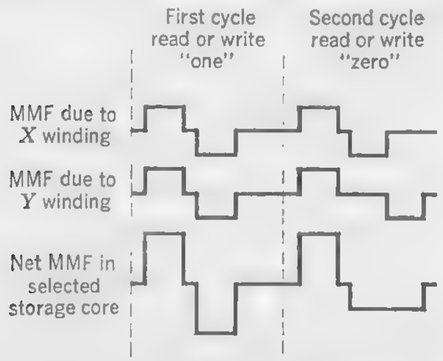


FIG. 12.51. Cycle when switch control is used to operate matrices in parallel.

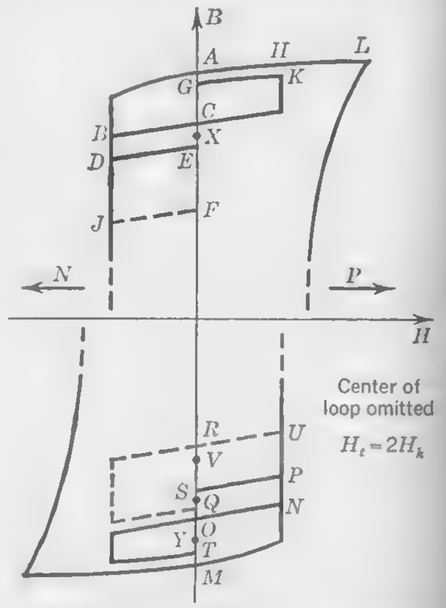


FIG. 12.52. Idealized minor loops associated with storage operation.

all the Y switches corresponding to 0's first, and then to restore the X switch and the remaining Y switches.

12.7.9. Behavior of Minor Loops in Storage Cores. In practice every operation performed on a storage core involves minor hysteresis loops. Only rarely is a core allowed to settle onto a portion of the major loop. For example, a core with a 1-μsec switching time is usually driven with pulses which may be as short as 1.2 or as long as 3 μsec in duration. Such a core is still in process of settling at the end of a pulse which is 40 μsec long. Even though the applied current may be sufficient to force the core onto the major loop, ordinarily the pulses would not be of sufficient duration to allow the core to settle completely. Considerable attention should be paid to the settling problem when extremely rectangular cores which have a high value of knee are used. With cores of this type, it is possible to vary the operating current over a wide range of values without apparently affecting the storage properties. However, the ultimate settling-time constant of a core is strongly dependent on the drive current. In designing storage units which employ very rectangular cores, it is necessary to conduct adequate experiments over the complete operating-current range to determine the settling characteristics of the cores.

In storage matrices the minor loops are extremely important. Figure 12.52 shows some typical but idealized minor loops associated with storage-core operation. To illustrate the problem, the worst possible storage loop in which $H_t = 2H_k$ has been chosen. Point A is what is usually defined as the 0 storage state. The 0 state set during the first part of a cycle is followed by at least one half excitation during the second part of a cycle. Therefore, the initial 0 state is point C rather than the desired point A. If a series of n-direction half excitations are applied, the core will progress through loops such as $CBDE$ until ultimately a point F is reached. At this point

any further half excitations will produce no more flux change, and the core will remain on the loop FJF. On the other hand, if subsequent half excitations are in the p direction, the core will move through loops such as CKG until a final stable loop AHA is attained. During a reading action the core is driven to L and then returned to A. Very little output occurs if loop ALA is traversed. In many storage units the action at the 0 state consists of a complex of p and n half excitations, and the actual loops tend to fall in the $ADEKH$ region.

To summarize these effects: Any disturbances caused by a writing or storage operation tend to move the core to a point such as C, E, or F; whereas disturbances in the read direction tend to move the core toward point A, which is the desired storage state. The lower half of Fig. 12.52 shows loops associated with the 1 state. In most storage cycles, point M is the original storage state. However, it is not so important for the core to remain at point M as it was to remain at point A, because demagnetization from M will result in a 1 signal of reduced amplitude while demagnetization from point A will result in a signal which is large instead of being of 0 or low amplitude. Excursions via loops MNO, $ONPQ$, etc., up to point R are often permitted or even encouraged in the design. As in the case of the 0 state, the actual loops in most storage units also include such loops as RS and OT.

The minor loops which are traversed in the 0 and 1 states alter the amount of signal obtained when a selected core is read, but this is not so important as the effect on the signals produced by half-excited cores. The sense winding is arranged to link an equal number of half-excited cores in the positive and negative directions. Consider the case of the positive cores being at point A and the negative cores at point M when the reading operation is performed. Since the flux change AH is small and the flux change MN is large, there will be imperfect cancellation. Furthermore, there are great differences between loops AHA and MNO. AHA is a saturated loop of low permeability and involves little or no domain-wall movement. The voltage output from such a loop is a sharp peak which settles rapidly. Loop MNO traverses a region of high permeability involving domain-wall movement, and the voltage output is a rounded peak with a long settling time. There are differences in the flux output of the various loops and in the shape of the voltage pulse. If a large matrix is imperfectly driven, these differences can prevent the reading of the signal from a selected core. Two methods of solving the driving problem are described below.

Use of Saturated Loops. This approach is feasible when the matrix has a digit-plane winding. A selected core is initially set in state C for 0 or state M for 1. Subsequently, a nonselected core can be subjected to the types of disturbance listed in Table 12.24.

Table 12.24. Types of Disturbance Applied to Selected and Nonselected Cores

Nonselected core is on	Selected core is set to	
	1 state	0 state
Selected line	Half-p, half-n action	Half-p action
Nonselected line	No action	Half-p action

The two possible types of excitation are a net half-p and a symmetrical half-p' half-n. Under normal conditions of operation there is an equal probability of writing a 0 or a 1 into a matrix. During a 1 action $2(n-1)$ cores are disturbed in an n^2 matrix. During a 0 action $n^2 - 1$ cores are disturbed. Therefore, the ratio of the probability of the half-p action to the probability of a half-p, half-n action is $(n^2 - 1)/2(n-1)$ or $(n+1)/2$. In most matrices this factor becomes large, as can be seen in Table 12.25.

The tendency is to favor the half-p excitations, which would move the cores onto loops such as AHA and RUR. Because these loops have similar characteristics,

nearly complete cancellation is obtained. The half-p, half-n type of disturbance tends to move the core from points A and R to points C and S. As a result of this, most cores in the matrix operate on non-domain-wall-movement loops, while the remaining cores operate on domain-wall-movement loops. This can lead to imperfect cancellation of the disturbed signals. In small matrices the resulting disturbance can be low enough to be ignored, but in large matrices this source of error can be dangerous. In order to correct the disturbance, all the digit-plane drivers can be operated for a short time after any storage cycle to produce what is known as a post-write disturb pulse.

Table 12.25. Relationship of Number of Cores in Matrix to Half-p/Half-p, Half-n Actions

Number of Cores in Matrix	*Number of Half-p Actions per Half-p, Half-n Action (Average)*
64	4.5
4,096	32.5
262,144	256.5
1,000,000	500.5

A post-write disturb pulse moves a core which has been left at C or S up to points such as G or V which are not too far removed from the correct storage point. A few more applications of pulse will bring the loops to A and R. An important advantage of this type of cycle is that very little domain-wall movement occurs in the disturbed signals. Therefore, the disturbances decay rapidly and it is possible to employ gating to improve the signal-noise ratio. The use of a post-write disturb pulse has other advantages. It maintains all cores on a loop in which the disturbance due either to the operation of the digit-plane driver or to the operation of any of the address drivers is at a minimum, and thereby essentially reduces the inductance of the whole system to a minimum value. If it is desired to drive extremely large arrays, the introduction of a post-write disturb pulse can simplify the driving problem considerably. The disadvantage of a post-write disturb pulse is that it increases the cycle time and also introduces noise into the cycle in the same way that the normal digit-plane pulse does. For this reason, a post-write disturb pulse operation is usually avoided if possible.

Unequal Currents in the X, Y, and Z Lines. Another solution to the driving problem is to use unequal currents in the X, Y, and Z lines of a matrix. Use of this technique requires careful experimentation to achieve the correct balance of currents. The disadvantage of the technique is that it always reduces the working-current tolerance of the core array; but if the cores have sufficiently high "knee," it is usually possible to maintain sufficient tolerance. Any one of a number of methods can be used to introduce this unbalance of current to a system. In practice, it is found that the performance of a memory can be improved both by increasing and by decreasing the value of the digit-plane current, although theoretically optimum results would be obtained by using a slightly high value of digit-plane current. Performance can also be improved by employing different values of positive and negative current in the X and Y directions if the drivers can be set to different currents in these two directions. Empirical results indicate that there is always an optimum method of compensating any particular memory device. The method depends partly on the design of the device and partly on the usual mode of operation of the device. The correct method is determined by experimentation on the first model of each memory device that is produced.

12.7.10. Linear or Word Selection. An exaggerated form of the unequal-drive-current operation is the linear or word-selection method of reading. The windings and cycles used in this method are illustrated in Fig. 12.53. Coincident current is not employed for reading; instead, as many drivers or a switch giving as many outputs as there are characters or words stored in the storage unit are used. Reading is accomplished by providing very large pulses to the read drive winding. The four turns of this winding indicate that the drive current is approximately twice as large

as the current normally required for coincident-current operation. Under these conditions the core turns over very rapidly, and a very large read signal is emitted if the core is in the 1 state. If the core is in the 0 state, a non-domain-wall-movement type of disturbance is produced which is short in duration and very small in amplitude. Because the sense winding links only the disturbed cores, extremely high signal-noise ratios are obtained with this type of operation.

Two writing methods may be used with linear-selection cores. The first is illustrated in waveform in Fig. 12.53 under Cycle Type A. In method A, the write drive current is half the required current to set the core to the 1 state. A digit-plane winding, wound in the same sense as the write drive winding, is also energized if a 1 is to be written. In the method shown as Cycle Type B, a write drive signal large enough to turn the core to the 1 state is applied at all times. The digit-plane winding, which opposes the write drive winding, is energized only when the core is to remain in the 0 state.

Method B exaggerates the non-domain-wall-movement disturbances since both the read and the digit-plane signals tend to force the cores onto an asymmetric loop. The read signal is so large that the loop employed is more analogous to a switch-core loop than to the normal storage-core loop, and the half-n disturbance produced by the write line has little, if any, effect on the storage position of the core. Very small 0 signals are therefore obtained. This particular method of operation is employed in those cycles where it is either desired to obtain very large read signals or where it is desired to operate cores on a shorter cycle than that normally defined by coincident-current operation. Method B has the additional advantage that the cores are extremely tolerant to current variation. All the drive and digit-plane signals may be varied through wide ranges without serious effects on the performance of the storage unit. Cores operated in this manner will tolerate a wider temperature range than is possible with normal coincident-current operation. They will also operate to a higher maximum temperature because the cycle permits the use of much less rectangular materials than are required for use in coincident-current operation.

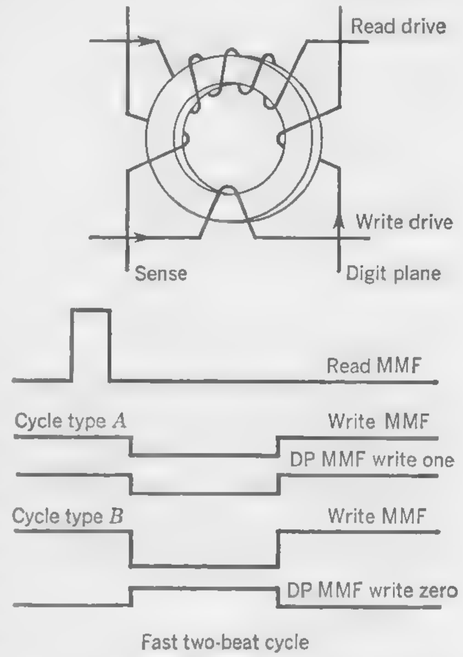


FIG. 12.53. Windings and cycle waveforms for linear selection.

Cores with normal rectangular loops may be operated by the methods described above with a considerable improvement in operation at the expense of additional electronic equipment. Special cores have been designed for use in the linear-selection mode. In these cores the hysteresis loop shown in Fig. 12.44 has been greatly exaggerated. Cores are made very small in diameter and wafer thin to obtain a large surface-volume ratio and, consequently, maximum heat transfer. Use of a material with a low maximum flux change reduces the heat generated within the core.

Electrically the core is designed to have very flat saturation characteristics. The knee on the left-hand side of the loop, if a normal rectangular-loop material is employed, is extremely abrupt and sharp. A knee value approaching 90 per cent is not difficult to obtain. Materials used are degraded considerably to reduce the knee down to the conventional 65 or 70 per cent. This allows much faster switching in the write portion of the cycle than can be obtained with conventional coincident-current material. On the right-hand side of the loop, where very small permeabilities are encountered even in rectangular-loop materials, it is important that minimum permeability be obtained. The core will then switch very rapidly when a large drive

pulse is applied during reading and will still switch rapidly during coincident-current write operations.

It is essential to use cycle *A* shown in Fig. 12.53 with cores designed for the linear-selection mode because the core materials have such poor rectangular-loop properties that they will not tolerate the disturbances introduced by cycle *B*. The properties of these cores are unique in that they can be driven with pulses having a very slow rise time. For example, a typical material will switch in 0.2 μsec if a pulse is applied which rises in 0.2 μsec and then falls at about the same rate so that the read pulse is completely triangular. The writing operation requires much lower currents and somewhat more rapid rise and fall times in order to obtain a pulse that is more nearly rectangular. With a rise time of about 0.1 μsec in the write mode, the core switches in about 0.3 μsec. With cores of this type, it is not difficult to obtain complete cycle times of less than 1 μsec, although transistor limitations tend to enforce cycle times somewhat larger than this for most of the large memories. At present the limitation on this technique appears to be one of availability of transistors. It is confidently expected that cycle times of a small fraction of a microsecond can be achieved within the next few years as better transistors become available to designers of core-storage units.

12.7.11. Matrices and Sense Windings. Matrices are usually assembled in a square or rectangular shape with the *X* and *Y* selection wires at right angles to each other. This is a matter of convenience rather than necessity, since a diamond-shaped matrix could easily be used. If rectangular matrices are used, the first departure from the square is usually selected. For example, a 4,096-bit matrix can be driven by two 64-way switches, which in turn can be driven by 32 drivers. It can also be driven by one 32-way switch and one 128-way switch, which can be driven by 36 drivers. Matrices can be folded so that the cores are in two layers, making a compact package in which one set of wires enters and leaves the matrix on the same side. This has some advantages in assembly and in the construction of some types of sense winding.

The *X* and *Y* wires are frequently made of copperweld, a wire of copper solidly welded around a steel center. This wire, more resistive and stronger than copper, simplifies assembly and produces a sturdy final product. The sense winding is always made of maximum-diameter copper wire. It is important to keep the resistive impedance of the sense winding as low as possible in order to effect the maximum power transference from the selected core to the reading amplifier. The digit-plane winding can be made of either copper or copperweld. Matrix wires are always coated with a heavy grade of tough insulating material. Thin insulating coatings would be destroyed by abrasion from the cores during assembly. After a matrix has been tested and assembled, it must be covered with a light flexible cement to glue the wires and cores into a solid mat. If this precaution is not taken, the wires and cores will chafe with any subsequent vibration, and breakdown of the insulation will occur.

Design of Sense Windings and Elimination of Disturbances. The *X*, *Y*, and digit-plane windings are relatively simple to design. The configuration of the sense winding dominates the design of the matrix. Electrical disturbances introduced into the sense winding as a result of the selection of the *X* and *Y* wires must be taken into account in the final mechanical design. The three types of disturbance introduced are as follows:

1. Disturbance due to the cores themselves.
2. Magnetic pickup which occurs because the sense winding and the *X* and *Y* wires are inductively coupled loops. This problem is solved by reducing the effective area of the sensing loops to a minimum.
3. Electrostatic pickup. When a number of matrices are driven in cascade, the instantaneous voltage of the *X* and *Y* wires may rise to 20 or more volts. Since the signal which is to be detected in the matrix is never more than 0.1 volt and since a 10-mv disturbance is undesirable, there must be considerable rejection between the *X* and *Y* wires and the reading amplifier to obtain adequate performance.

A correctly designed matrix minimizes the effect of all three types of disturbance. Cancellation of disturbances is complicated by the fact that there is a considerable

propagation time in the sense windings of large matrices. For example, the propagation time is usually between 0.1 and 0.25 μsec in a 4,096-bit array. In very large matrices the sense winding must be broken up into a number of sections so that the propagation time does not become too long. If the storage cores are widely spaced in the matrix, the propagation time in the sense winding will be long. In addition, the inductance of the address wires and the digit wire will be large; and this, in a matrix of any reasonable size, will introduce difficulties in driving the cores. In a well-designed matrix, about half the load presented to a driver is due to the air inductance of the winding and the other half to the nonlinear inductance of the cores. This approximation is based on the assumption that the cores in the matrix are as close together as possible. If the matrices are to be driven at a maximum rate, it is highly desirable to pack the cores so that they are almost touching. This particular point of minimum size in a matrix presents a considerable problem to the designer. He can improve the electrical design by using the maximum packing density, but at the same time he will inevitably increase the manufacturing cost of the matrix. In recent years improved manufacturing techniques have permitted designers to employ spacing between matrix wires which is nearly ideal. In most present-day matrices, the distance between adjacent wires is a little more than the outside diameter of the core that is used in the matrix. Figure 12.54 is a diagram of a recently designed matrix of maximum density.

Disturbance Due to Cores. Minor loop disturbances due to the selected *X* and *Y* wires have been discussed previously. Disturbances of a secondary order on a large number of other *X* and *Y* lines also arise because the switches are not perfect. The sense winding must be wound in such a manner that cancellation occurs repeatedly over very short sections of the winding.

Magnetic Pickup. The sense winding must be wound so that its net area is zero or nearly zero. This means that the input and output must be at the same point. Furthermore, any wire which has a component of direction parallel to the *X* or *Y* wires must have in close proximity to it a wire which travels in the opposite direction and which has the same components with regard to the *X* and *Y* wires. The process is somewhat analogous to the methods used in noninductively winding high-quality wirewound resistors and register cards.

Electrostatic Pickup. This is one of the most troublesome of all types of disturbance. Because the capacitance between the sense winding and the *X* and *Y* wires is large, it is difficult to prevent the sense winding from moving with the *X* and *Y* wires. A correctly designed sense winding moves in such a fashion that no net voltage appears across the two sense-winding output wires even though the whole sense winding may move through a considerable excursion. This must be true regardless of which *X* and *Y* wires are selected. Information is coupled out of the sense winding either by means of a difference amplifier or by a balanced transformer. Investigation has shown that most of the disturbance in a matrix is due to electrostatic pickup. Improvements in the 1/0 ratio by as much as 3 or 4 to 1 have been achieved by apparently minor alterations to the sense winding. The best solutions to this problem have been obtained by balancing the electrostatic pickup in a symmetrical manner around the center of the winding. This balance must be maintained for all combinations of *X* and *Y* selections.

The principles of operation for two popular types of sense winding are described below:

Sense Winding—X and Y Pulses Are Simultaneous. Figure 12.55 is a diagram of a small sense winding of this type. The arrows on the *X* and *Y* lines show the manner in which these wires are driven with respect to each other to produce the same action. The signs against the cores show the polarity of the linkages of the sense winding. Moving from the center of the sense winding toward either *A* or *B*, the same pattern of polarities is encountered and the winding is symmetrical about the center. Consider that line X_0 is excited. Moving from *C* toward *A*, the excited linkages induce $-v_s$, $+v_s$ and cancellation occurs. Moving from *C* toward *B*, the same pattern is encountered. Consider that line Y_0 is excited. Moving from *C* to *A* or from *C* to *B*, the excited linkages induce $+v_s$, $-v_s$ and cancellation occurs. Complete analysis

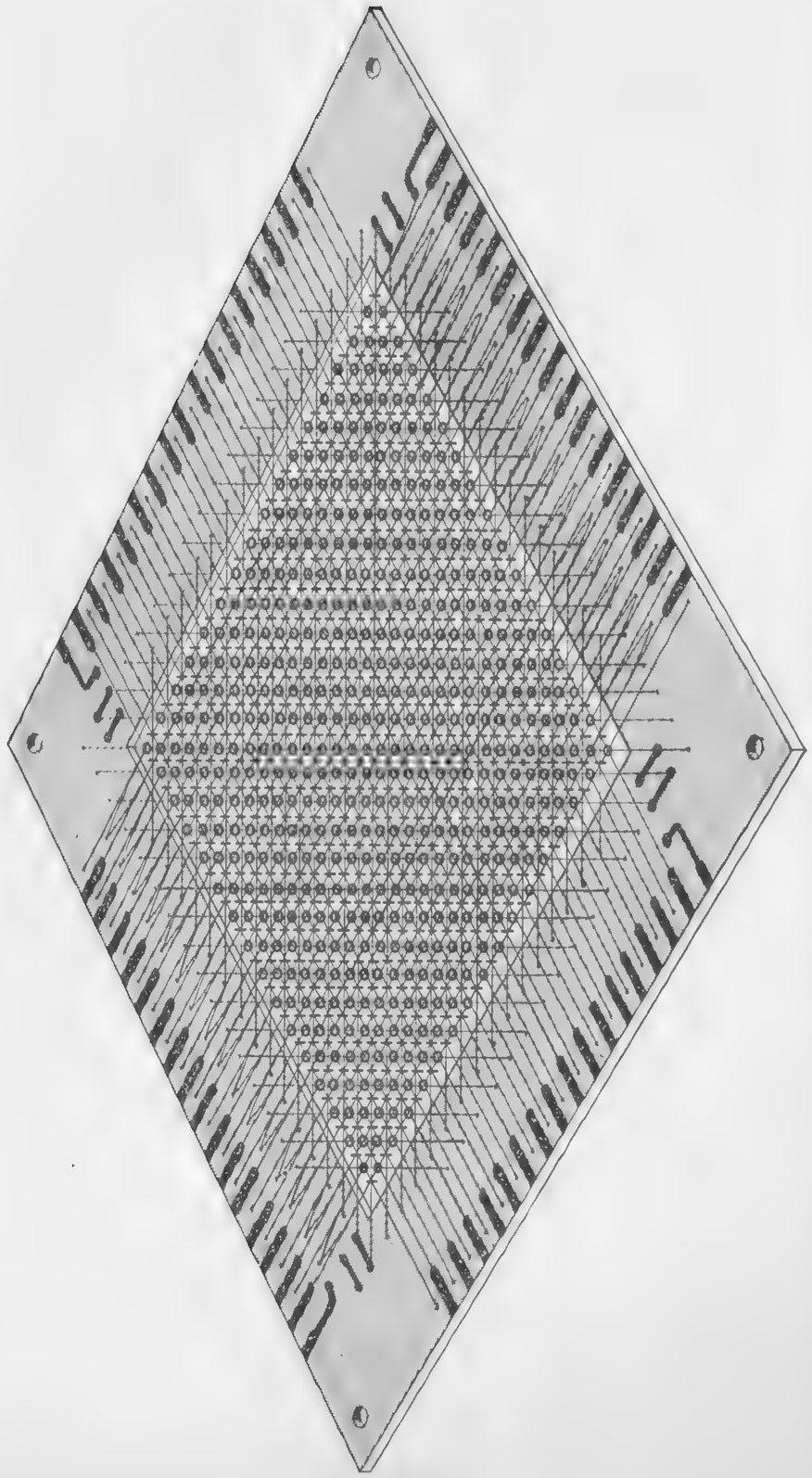


FIG. 12.54. Typical closely packed core matrix.

of the sense winding carried out by this method will indicate that cancellation is always adequate.

Sense Winding—X Pulse Commences before Y Pulse. Sometimes a high 1/0 ratio is required, and slightly slower operation may be permitted to achieve this improvement. In such cases the X pulse commences about 1 μsec ahead of the Y pulse. The sense winding is designed to minimize the disturbance due to the Y pulse at the

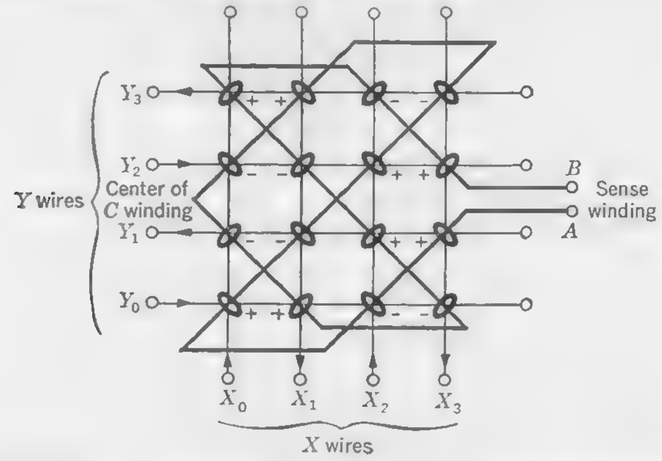


FIG. 12.55. Sense winding used when X and Y pulses are simultaneous.

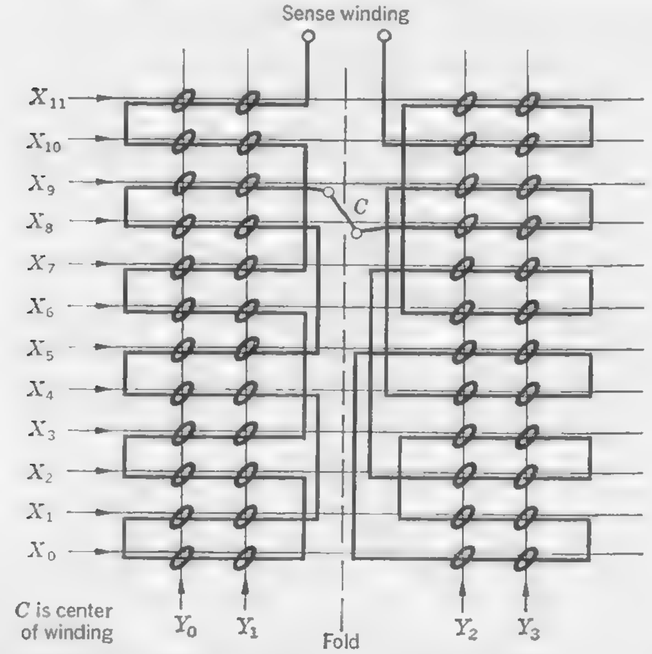


FIG. 12.56. Sense winding used when X pulse commences before Y pulse.

expense of increased disturbance due to the X pulse. The reading signal is gated and does not commence until just after the Y pulse has commenced. Use of this technique with a sense winding has made it possible to achieve 1/0 ratios of 30:1 in large matrices.

Figure 12.56 is a diagram of a small winding shown as a flat plane. This winding is usually employed in folded rectangular matrices. Y disturbances in the winding are canceled out every time the sense winding links a Y line for the second time. X disturbances can be canceled by bucking one half of the X line against the other

half. There is a minimum of electrostatic pickup with respect to the Y line and a maximum with respect to the X line. The winding has mechanical advantages in that the sense winding is parallel to the X line and is therefore easy to thread through the cores.

Characteristics of the Sense-winding Output Signal. Figure 12.57 shows some characteristic waveforms from a single core. The disturb signals are canceled in the sense winding, but each signal is delayed slightly with respect to the next. The

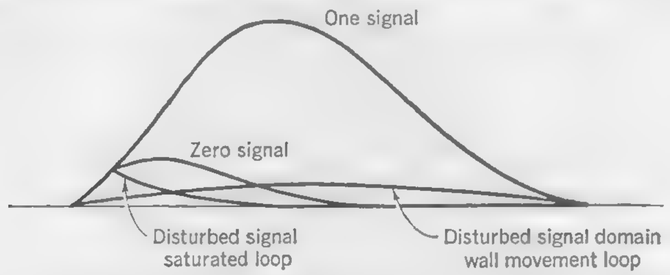


FIG. 12.57. Characteristic waveforms of output signals of a single core.

result is that complete cancellation does not occur even with a very good sense winding. The same type of delayed action occurs in magnetic and electrostatic pickup. The resulting output signal contains the following components:

1. A large-voltage excursion of the whole winding due to electrostatic pickup
2. A 1 or 0 signal plus any uncanceled disturbed signals on an area basis
3. A burst of high frequency of amplitude much greater than the 1 signal due to lack of cancellation of the disturbances in time

Component 1 can be rejected by a difference amplifier or by a transformer or, in some cases, by both. Component 2 is the actual reading signal which must be utilized by the equipment component. Component 3 can be rejected by means of a simple low-pass filter. This filter is usually arranged to cut off at a frequency which is just high enough to allow reasonably small distortion of the 1 signal.

Switches and Switching Techniques

12.7.12. Principles of Magnetic-switch Operation. When magnetic-core switches are used to drive the selection wires of a coincident-current matrix, the switches are usually arranged as small matrices which, in turn, are driven by an anticoincident form of drive rather than the coincident-current form. Consider the circuit shown in Fig. 12.58 and the operating loop shown in Fig. 12.59. A current producing mmf AD

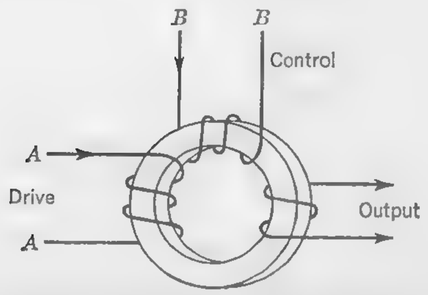


FIG. 12.58. Magnetic-core-switch circuit.

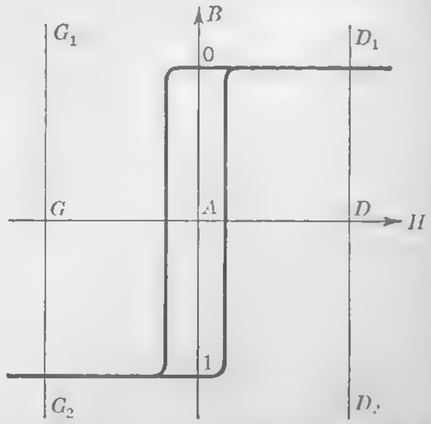


FIG. 12.59. Operating loop for magnetic-core switch shown in Fig. 12.58.

is applied to drive winding AA. A control current producing mmf AG is applied to the control winding BB. Let $GA = AD$ and let the core be in the 1 state. If the control winding is energized before the drive winding, there will be very little flux change and virtually no output will appear. If only the drive winding is energized, there will be a large flux change and the mmf, which is included between the line D_1DD_2 and the hysteresis loop, will be available to the load at the output winding. The core is reset by reversing the current in the drive winding to produce mmf AG. An output which is of opposite polarity to the original output will be produced.

A matrix of switch cores may be wound so that one of $m \cdot n$ outputs is operated by the simultaneous driving of one of m wires and $n - 1$ of n wires. This principle can be extended to any number of coincidences. Table 12.26 indicates the number

Table 12.26. Number of Drivers for Various Matrix Sizes

No. of cores in the matrix	Double-coincidence matrix			Triple-coincidence matrix		
	No. of switches	Coincidence in switch		No. of switches	Coincidence in switch	
		Double	Triple		Double	Triple
64	16	12	12	12	12	Not possible
4,096	128	32	24	48	24	24
262,144	1,024	96	48	192	48	36
1,000,000	2,000	130	60	300	60	42

of drivers required for various matrix sizes when all the coordinates of the matrix are driven by switches.

The following facts may be noted from this table:

1. A double-coincidence matrix driven by triple-coincidence switches requires the same number of drivers as a triple-coincidence matrix driven by double-coincidence switches.
2. The triple-coincidence matrix driven by triple-coincidence switches does not employ many less drivers than a double-coincidence matrix driven by triple-coincidence switches.

12.7.13. Magnetic-core Switches. Magnetic-core switches provide a means of driving the large number of selection wires associated with the storage matrices from a reasonable number of electronic drivers. They replace what would otherwise consist of two drivers per matrix line plus a large tree of diode logic used for decoding purposes. In this sense they represent the first use of cores as a logical element. Switches have been designed to replace diode adders and to perform other logical functions; they are likely to find extensive logical use when computers are designed around the use of magnetic components.

A simple switch and the cores associated with it act as a transformer. Consider, for example, a multiple-winding transformer which has a core of rectangular material. There are two primary windings and one secondary winding. A small signal may be applied to one of the primary windings and a control current to the other. When the control winding maintains the core in the saturated region of the operating loop, there will be virtually no output in the secondary winding. As the control current is altered, there will be no output due to the signal until a signal–control-current relationship has been established which allows the signal to operate beyond the knee of the curve into the high-permeability region. There will be an output as the control current is changed because the saturation characteristics are not completely flat. To prevent this unwanted secondary signal, a switch-core material having a ratio of remanent flux density B_r to saturated flux density B_s as close to unity as possible is most desirable. Materials having a B_r/B_s ratio of less than 0.9 are not

usually desirable; the best materials have a ratio of at least 0.95. When the signal is in the region of very high permeability, the transformer is almost perfect in its characteristics; but when it is in the region of decreasing permeability, the transformer becomes more and more imperfect until the signal is again in the saturated region. In the best switch-core materials the amount of total flux change in the region of decreasing permeability is small and the rate of change of permeability is very large. For a given flux output, the total area of the loop should be as small as possible to keep eddy-current and other dynamic losses to a minimum. This is particularly important in switch cores because their average power transmission is relatively large.

Switches are normally driven by flat-topped current pulses. Switch cores in the normal state are in a saturated region such as state 1 in Fig. 12.59. If the net ampere-turns applied to the switch core leave it at state 1 or take it toward G_2, the switch core will not produce an output. On the other hand, an output is produced if the net ampere-turns take the core beyond the knee of the curve toward D_2.

When the switch core is in the operating state, it acts as a current transformer in which the output current is equal to the current available between line D_1DD_2 and the hysteresis loop. The magnetizing ampere-turns vary between H_k and H_t as the switch operates. If H_i is the input ampere-turns and H_o is the output ampere-turns, they can be expressed in terms of limits such that $H_i - H_t \leq H_o \leq H_i - H_k$.

In most cases it is desired to achieve a pulse of a given "flatness" from the output winding. As an example, let $H_t = 1.75H_k$ and let the output pulse droop by 5 per cent at the top. Let $H_i = nH_k$. Then

$$nH_k - 1.75H_k \leq H_o \leq nH_k - H_k$$
$$(n - 1.75)H_k \leq H_o \leq (n - 1)H_k$$

but H_o at the lower limit is 0.95 times its value at the upper limit. Hence $n = 16$. To achieve this high value of n, the number of ampere-turns on both primary and secondary windings would be large, and as a direct corollary, the number of turns would be large. This would tend to produce a physically large and inductive switch which would be difficult to drive. Therefore, switches are often operated over only a part of the loop in order to keep the variation of magnetizing current to a minimum. In most core material $H_c = 1.1H_k$, and if the core is used over half its characteristic, then $n = 3$. Switches are rarely constructed with n greater than 6 and have been constructed of exceptionally good materials with n as low as 2.

A driver which is driving a magnetic switch has as its load one switch core which is acting as a transformer and a number of other switch cores which are in a saturated state. The driver must maintain constant-current conditions into this inductive load. Consider the effect of the saturated cores alone since in most cases the voltage drop due to the operating core is trivial. If i is the driver peak current, N is the number of primary turns, k is the factor which depends on the time of rise of the current, on the cross-section area of the switch core, and on the saturated permeability of the loop; and if there are m saturated switch cores, then the voltage drop V will be kiN^2m. For a given output the ampere-turns iN must be constant. Therefore, any attempt to reduce i is accompanied by a rapid increase in V. H_t is proportional to iN. H_k for a given core is fixed. Therefore, if it is desired to increase n, then iN must be increased. The following relationships govern the design of switches once a core size and material have been chosen:

$$V \propto \frac{1}{i} \quad \text{when } n \text{ is constant}$$
$$V \propto N^2 \quad \text{when } i \text{ and } n \text{ are constant}$$
$$V \propto n^2 \quad \text{when } i \text{ is constant}$$

Actually these equations do not imply the complete relationships. In order to keep the voltage drop in a switch low, the cores of minimum diameter are employed to keep the magnetizing current to a minimum. An increase in N in most cases represents an increase in the size of the core and a constant increase in the desired iN.

Therefore, if the core size is also considered, the above relationships are altered. For normal switch sizes the following relationships are true on an empirical basis:

$$V \propto \frac{1}{i^x} \quad \text{when } n \text{ is constant and } 1.5 < x < 2$$
$$V \propto N^3 \quad \text{when } i \text{ and } n \text{ are constant}$$
$$V \propto n^3 \quad \text{when } i \text{ is constant}$$

Designing a switch involves a difficult compromise between adequate performance and reasonable driving voltage and current.

An ever-present problem in system design is how many coincidences may be utilized in a switch. The initial tendency is to choose the design which requires the least number of drivers. However, a reduction in the number of drivers necessitates a disproportionate increase in the size of the drivers and, consequently, system efficiency decreases.

Switch wiring diagrams can become so complex that it is necessary to employ special symbols to describe them. Figure 12.60 shows the basic symbols and defines

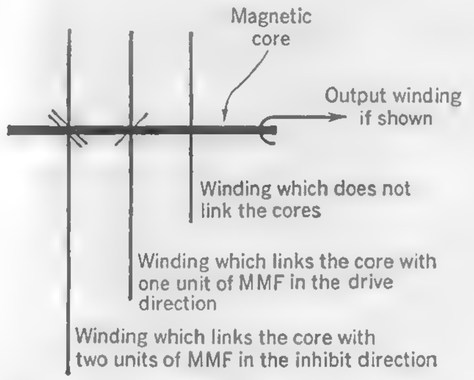

Magnetic core
Output winding if shown
Winding which does not link the cores
Winding which links the core with one unit of MMF in the drive direction
Winding which links the core with two units of MMF in the inhibit direction

FIG. 12.60. Symbols used in switch wiring diagrams.

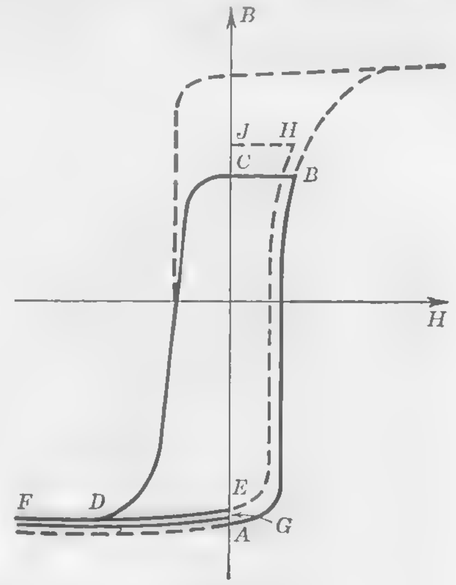


FIG. 12.61. Typical action of a switch core on a hysteresis loop.

their meaning. The output windings are frequently omitted in the diagrams because, except in unusual circumstances, every switch core provides one output.

Behavior of Minor Loops in Switch Cores. Figure 12.61 shows a typical action in a switch core. The core is initially at point A, which is on the major loop. It is then driven to point B by means of a drive pulse. The drive is removed, and the core settles to point C. A reverse drive is then applied which is removed at point D, and the core settles to point E. In a normal switch there is a high probability that the core will receive subsequent reverse drive pulses which produce loops such as EFG and may ultimately settle to A, but the next action after the core is set at E may be a drive action. During this action the core is on a minor loop which is narrower than the original loop and a little more current will be available to the load, resulting in a slightly higher flux transference from the core to the load. The core will move to H and, on release of the drive, to J. The flux CJ will be greater than the flux AE. This action is cumulative, and the operating loop will move until it is centered around the origin. In practice the core receives a mixed sequence of forward and reverse drive pulses; its behavior depends upon the immediate past history. Because the core drifts from giving a relatively small output to a relatively large output, extremely poor operation of the storage matrices results. To overcome this, the reverse drive imposed from C to D is usually maintained for a longer time than the drive from A to B. The core will still move from the major loop, but by only a small amount. The exact amount by which the duration of the reverse drive pulse should exceed the duration of the

drive pulse is a matter of experiment since it is a function of material, load, and drive characteristics. In some systems all the switch-inhibit drivers are operated during some portion of the cycle with the same effect as the lengthened reset pulse.

An ideal method for minimizing loop drift is to ensure that the switch core is always driven from one point of saturation to another. Switches which employ operation from one state of saturation to another usually require auxiliary equipment for stabilizing either the voltage output from the switch or the current output from the switch or, in some cases, both.

Binary Switches. Binary switches are probably the most elegant of all switch designs. An eight-way version is shown in Fig. 12.62. The switch decodes a binary number which is expressed as 2^q, 2^{q-1}, . . . , -2^0 into 2^{q+1} combinations. The switch is driven by $q + 1$ pairs of drivers, in which one driver of the pair operates for a 0 and the other for a 1. A drive winding provides one unit of mmf and an inhibit winding q units. All activated drivers combine to provide the drive current. Restoration is effected by a separate restoration winding which provides $q + 1$ units of mmf. The selected core has $q + 1$ units of drive. Those cores whose address differs by one

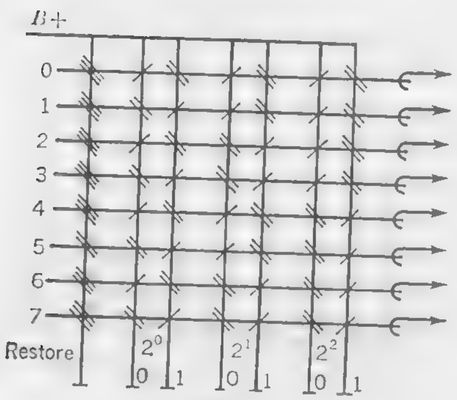


FIG. 12.62. Eight-way binary switch.

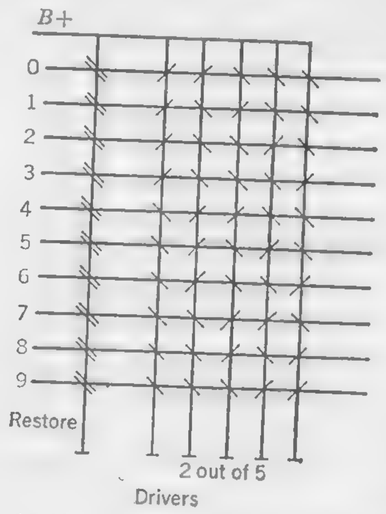


FIG. 12.63. Two-out-of-five code binary switch.

digit from the selected number have 0 drive and those whose address differs by two digits have $-(q + 1)$ units. Therefore, only the selected core provides an output.

Each core is linked by a number of wires proportional to $(q + 1) \cdot (q + 2)$ and provides $q + 1$ output. The number of turns per core and the load on each driver rise very rapidly as q is increased. The switch relies on coincidence of the driving currents at all points in the switch; but this is difficult to achieve because each driving line acts as a delay line in which the delay at any point in the line is a function of the number of turns up to that point. For these reasons, the switch is useful only in small sizes.

The binary switch has the unusual feature that all the activated drivers combine to provide the output. With certain types of redundant code this feature can lead to improved operation. An example is the two-out-of-five code switch illustrated in Fig. 12.63. In this switch only those drivers corresponding to a 1 are driven. It is also possible to design single-ended binary switches, but these are less efficient than the push-pull version described above.

Switches which fully utilize the combinatorial advantages of cores are almost useless in storage units, but they provide the basis on which functions such as addition can be mechanized. In such applications there is no requirement that the switch should be efficient or that the output waveform should be well shaped. Quite large switches can be used to perform complicated functions.

Switches in Which Only One Driver Contributes to the Output Current. Consider a core which has four primary windings and one secondary winding. Let all the primary windings be fed with equal numbers of ampere-turns. Let one winding A be in the drive direction and the other windings B, C, and D be in the inhibit direction. Then an output is obtained if A, and not B or C or D. This operation is fundamental to all switches of this type. The only amplitude requirement is that the mmf fed to B, C, or D is never less than that fed to A. Because of this type of operation, these switches are often called anticoincidence switches.

Figure 12.64 shows an eight-way binary switch of this type. If the same terms are used as were employed in the previous binary switch, then there are 2^{q+1} outputs as a result of $2(q + 1)$ drivers. The 2^0 driver pair acts as the drivers; the others act as inhibitors. The drive to the selected core is one unit, to the next selected is zero, and to the remaining is negative. Restoration is effected by causing both drivers of one of the inhibit driver pairs to conduct. A separate restoration winding can be used if desired. The number of turns per core is proportional to $q + 1$ and the output is 1. In this respect, the switch utilizes wire no more efficiently than did the original binary switch.

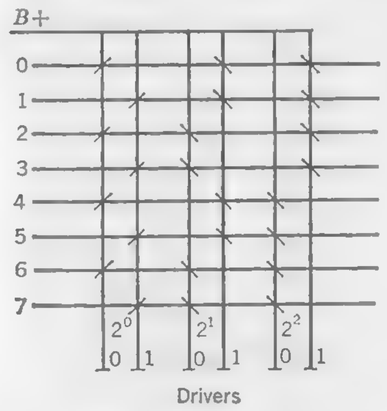


FIG. 12.64. Eight-way anticoincident binary switch.

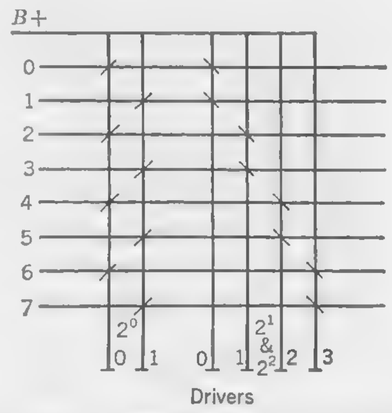


FIG. 12.65. Eight-way triple-coincidence switch.

There is no difficulty in achieving coincidence in a large switch of this type. In practice, double coincidence or, at most, triple coincidence is employed. An eight-way switch of this type is illustrated in Fig. 12.65. In this case, the 2^1 and 2^2 bits are decoded; and the three drivers which are not selected, as a result of decoding, are allowed to conduct.

The number of wires per core increases with switch size. The number of ampere-turns driven by a driver increases as $2^{\frac{q+1}{2}}$, which is an improvement on the pure binary switches. If there is triple coincidence, then the ampere-turns driven by a driver increases as $2^{\frac{2}{3}(q+1)}$. As a generalization the driven ampere-turns for a n-way coincidence increase as $2^{\frac{n-1}{n}(q+1)}$. It can be seen in Table 12.27 that as n increases the switch approaches the efficiency of the binary switch rather rapidly. Therefore, double coincidence is usually employed.

In order to obtain the relative driver size, the largest load in the triple-coincidence switch was compared with the largest load in the double-coincidence switch. It should be noted that the number of drivers used for triple coincidence rises rather slowly with respect to the number of drivers for pure binary, while the number of drivers for double coincidence rises rapidly. In most cases the total power required is greater in triple-coincidence operation than in double-coincidence operation. Certain switches which are a cube, such as 512 or 1,000, are efficient under triple-coincidence conditions.

It is possible to construct and operate switches of this type on somewhat different principles. A winding links all cores of the switch in the inhibit direction. This winding is d-c driven by a precise, high-impedance, constant-current driver. Usually the winding has few turns and the current flowing in it is high. In an $n - 1$ coincidence switch the ampere-turns provided by this winding to each core must be exactly $n - 1$ times the ampere-turns provided by any selection driver. The selection windings link the switch in a conventional manner except that they are all in the drive direction. One driver is operated in each set to select the switch core. For example, in a double-coincidence 64-way switch there would be two sets of eight drivers and the d-c constant-current driver. If A is the ampere-turns provided by any driver, then initially the drive to all the cores will be $-A$. The two selected drivers will provide a total of $2A$. The resultant drive to the selected core is $+A$. The drive to the cores on the selected lines is zero and to all other cores is $-A$.

Table 12.27. Number of Drivers for Binary, Double-coincidence, and Triple-coincidence Switch Operation

No. of positions in the switch	No. of drivers for binary	No. of drivers of relative size 1 for double coincidence	No. of drivers for triple coincidence	Relative driver size	Drivers times size
16	8	8	8	1	8
64	12	16	12	2	24
128	14	24	16	2	32
256	16	32	20	4	80
512	18	48	24	2	48
1,024	20	64	32	4	128

To operate the switch both selected drivers are energized simultaneously and then a reset pulse is applied to all the cores by removing both drivers together. The switch has the advantage that the logic required to control it is simple. It has the following disadvantages:

1. The direct current flowing in the common winding must not change when the selection drivers are operated. If it does change, the winding acts as a shorted turn and removes mmf from the drive. Therefore, the constant-current source to this winding must be of high impedance. In addition, the voltage induced in the winding by the selection action must be small. If it is not, the capacitance associated with the winding will reduce the driving impedance. Therefore, the winding itself must be a low-impedance high-current winding, driven from a high-impedance low-capacitance source.

2. The direct current flowing in the common winding provides the reset mmf. If A is the mmf applied to the core on set, then a value B, which is greater than A, must be applied on reset, because the switch core moves from H_k to H_c on set and from H_c to H_t on reset. Therefore, the direct current must be adjusted to the value B, and the drivers adjusted to a value of $\frac{1}{2}(A + B)$.

3. The tolerance of the selection and bias currents must be small because the three currents are involved in any set action. In a normal switch only one current is involved in a set action. The drivers employed to drive this switch must be much more precise than those required in a normal switch. In practice it is usually found that only one-half or one-third of the tolerance is permitted for drivers used in conjunction with switches of this type.

4. Leading and trailing edges of the drive pulses must be controlled with care. The leading edge defines the set pulse, while the trailing edge defines the reset pulse. Therefore, the drivers must be specially designed to provide the correct pulse shape. In a normal switch the leading edge is controlled, but the trailing edge is simply allowed to fall smoothly without any particular control.

This switch has another distinct advantage apart from its logical simplicity. Because the switch cores are reset at all times except during an actual set action, the tendency toward an operating loop shift is small. This is particularly important during very fast cycles. A short reset action will occur at some time in every cycle, which tends to stabilize the operation of the switch.

Unidirectional Switches—Diodes in the Secondary Circuit. The magnetic switches described above produce bidirectional signals at the output, effect impedance matching, and introduce some combinatorial advantage. The last two features are always advantageous, but the production of bidirectional signals can be as much of a disadvantage as it is an advantage. If a bidirectional signal is produced from a normal switch, the cycle and mode of operation must be based on a two-beat principle, which may be an undesirable restriction. For this reason, there has been considerable interest in producing unidirectional switches.

One way of obtaining a unidirectional output is to drive the switch with a large current in order to obtain a pulse in the output winding and then to reset the switch very slowly so that essentially no current is obtained from the output winding. This mode of operation has been used successfully in a number of storage devices. In some devices the current produced at the secondary of the switch due to resetting the switch core has been maintained as significantly less than half the switching current in the matrices. During reset a current flows in the matrices, but it is too small to affect the stored information. A considerable time is expended in resetting the switches in this type of operation; therefore, it is applicable only to memory units in which the cycle time is long.

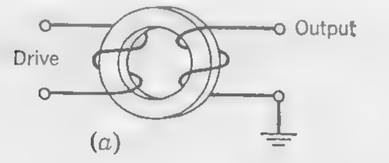
A second method of using conventional switches to provide an effectively unidirectional output is to reduce the reset drive to a point where the current from the secondary during reset is about half the value of the current from the secondary during set. This mode of operation has been used successfully to implement the drive operation in linear-selection or word-selection memory devices. Memory devices of this type frequently require about half the driving current during the write part of the cycle. This mode of switch operation will produce a large short pulse during the set operation and a long small-amplitude pulse during reset.

Another way of obtaining undirectional operation of a switch is to insert a diode in each secondary circuit. Under these conditions, current can flow from the switch core into the matrix in one direction only. The diode is highly reliable in this mode of operation. The voltages occurring in the secondary of the switch core can be very low and the diode does not require a large back resistance at any time in its life. The introduction of an electronic element will, of course, reduce the reliability of the magnetics; but it has been found that diodes used in this mode can be considered almost as reliable as the cores themselves.

A switch utilizing diodes in the secondaries has unique properties. Power will be transferred from the primary winding to the secondary winding when the diode is able to conduct; but if the primary current is in a direction such that the diode in the secondary cannot conduct, power cannot be transferred to the secondary winding, and all the power in the primary must be used in switching the core and overcoming losses in the core. Therefore, the introduction of the diode introduces properties beyond that of unidirectional operation of the switch. It is possible to set the core, using very low primary power and no power output from the secondary, and later to reset the core, transferring power from the primary to the secondary. The converse operation is also possible: power can be transferred from primary to secondary during the set operation, and later the core can be reset, using low primary power, without loss of speed. Unidirectional operation implies loss of time somewhere in the system because it is necessary either to set or to reset the core without output. In most systems using unidirectional operation, an overlapping type of cycle is adopted so that one core is being set or reset while another core is performing the opposite function. By this means time loss is limited to the small amount of time in setting the switch into an initial state or clearing it into a final state at the beginning and end of long major operations.

The basic operation of an element of a unidirectional switch is illustrated in Fig.

12.66. A simple switch, shown in Fig. 12.66*a*, consists of a transformer using a nonlinear core. A single drive winding has been shown, but ordinarily several windings are used in order to obtain combinatorial advantage. The basic element of the unidirectional switch is shown in Fig. 12.66*b*. This consists of the same transformer with the addition of a diode in the output circuit and, in series with the output circuit, a constant-current load. The output circuit is shown connected to a terminal T, because in switches of this type a constant-current load with an associated diode can be used in common to all elements of the system. Normally, current flows through the constant-current load through diode D_1 to ground. If a drive current is applied



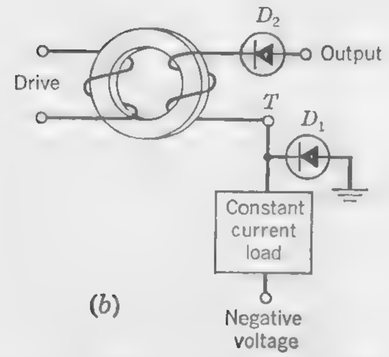
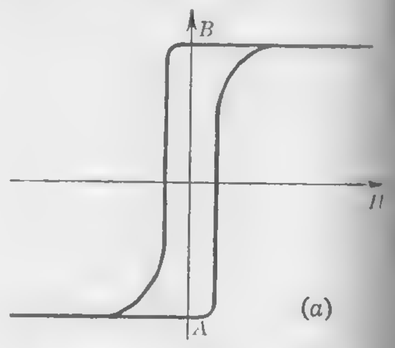


Fig. 12.66. (*a*) Simple transformer switch core. (*b*) Basic element of a unidirectional switch.



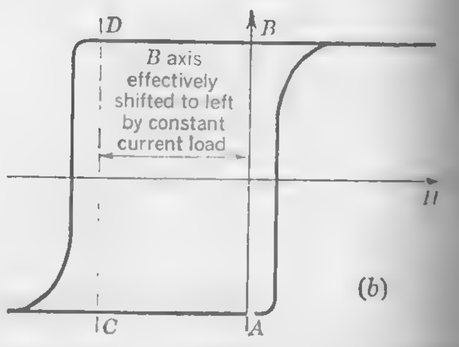


Fig. 12.67. (*a*) Hysteresis loop of simple switch core shown in Fig. 12.66*a*. (*b*) Effective loop used by unidirectional switch shown in Fig. 12.66*b*.

to the core to set it so that diode D_2 does not conduct, no current flows in the secondary circuit and a small amount of power is used in setting the switch core. If the core is now reset by applying a large current of the opposite polarity to the drive winding, diode D_2 can conduct. Current through this diode must be drawn from terminal T. It is usually drawn as the result of the secondary winding lifting terminal T above ground so that diode D_1 is cut off and the constant-current load is inserted as part of the load of the secondary of the switch core. This has the effect of defining the current that may flow in the secondary of this core precisely.

The addition of the diode to the secondary circuit has created a circuit in which unidirectional current flows at the output, in which very low power is required either to set or reset switch cores, and in which a constant-current load may be inserted in the secondary so that the current flowing in the output circuit is regulated to a high degree. This last feature is important in designing core memories. A switch of this type allows the designer to attain large combinatorial advantages and, at the same time, to route extremely precise currents into the storage elements themselves.

The BH loops describing the operation of the core are shown in Fig. 12.67. Loop *a*, the conventional square loop used by a switch core, defines the operation of the circuit shown in Fig. 12.66*a*. If the current corresponding to H is greater than is required to satisfy the loop, the excess current during either set or reset is transferred into the output winding. The loop used by the circuit in Fig. 12.66*b* is shown in Fig. 12.67*b*. During the set action, in which the core moves from point A to point B, the current

supplied by the primary circuit need be only enough to move the core through this excursion. It is not possible to transfer any excess current into the output circuit; and if excess primary current is provided, it will result in additional core losses and more rapid transition from A to B. During reset, however, the amount of current BD or AC must be provided before adequate current is available for switching the core. This current is defined by the constant-current load. The current provided to the primary will be distributed as BD into the secondary circuit, and any excess current above this amount will be used to switch the core and also to overcome losses in the core. A small amount of excess primary current will produce a much faster switching action of the core.

Because the constant current in the secondary is unidirectional, the constant-current load may feed a large number of secondary output circuits, provided that only one of these circuits is selected at a time. Under these conditions, the current flowing in the matrix is independent of switch characteristics. An eight-way switch operated with such a current arrangement is shown in Fig. 12.68. The switch is set by any of the conventional methods to store the address in the selected core. No power is transferred from the set winding to the output winding because the diodes block any flow of current. The current required to set the switch need be only sufficient to provide the customary 0 ampere-turns. Therefore, the inductance of the windings may be very low, and comparatively low-powered circuits may be used as set drivers.

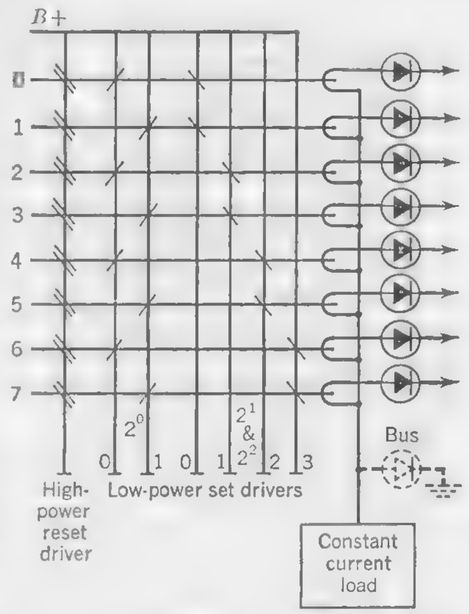


Fig. 12.68. Eight-way unidirectional switch.

When it is required to produce an output from the switch, the single high-power reset driver is operated and the selected core is restored to its normal state. The constant-current load may be either a pulsed or a d-c source; if it is the latter, a diode may be connected to the bus as illustrated to carry the current in the quiescent state. When the selected core is operated, the bus is depressed to cut off all the unselected diodes and current flows only in the selected line. It should be noted that the reset driver is really a constant-current load which absorbs power from the selected core. While the driver is supplying current to the output circuit, the core acts as a transformer. In large switches the extra diode on the bus is not required because the quiescent current is shared equally by all the output lines. The current in each line will be so small that it will introduce no adverse effects on the performance of the storage unit. Before deciding whether to use a diode, it is important to perform adequate end-of-life computations on the behavior of the diodes in the switch. It is possible for all the diodes to have deteriorated considerably in regard to forward resistance and for one of the diodes to have failed and been replaced by a new diode. This worst-case computation should be performed to ensure that a large proportion of the constant current is not being routed through one of the matrix lines.

The current relationships in the switch are rather complex because the cores are essentially current transformers feeding into a constant-current load. The mmf supplied by the reset winding must be equal to the output ampere-turns plus the magnetizing mmf plus an mmf absorbed as core losses. If either the driving current or the constant current alters, the new difference must be taken up almost entirely as an alteration of the core losses. In a highly efficient transformer, the losses are low and a small change in either of the currents may produce a large change in the losses. Losses are always associated with the switching time of the core. A small

increase in driving current can decrease the switching time and increase the voltage output. The excess voltage is taken up by the bus, but the reduced length of the output pulse presents a serious problem. Consequently, the two drivers must be very precise and, if they change, must change together. There is an effective way of solving this problem in small switches. The reset driver lines are connected to the constant-current bus in place of the constant-current driver. The drive current is then equal to the matrix current, the difference between driver and output ampere-turns is fixed, and the switch is stable.

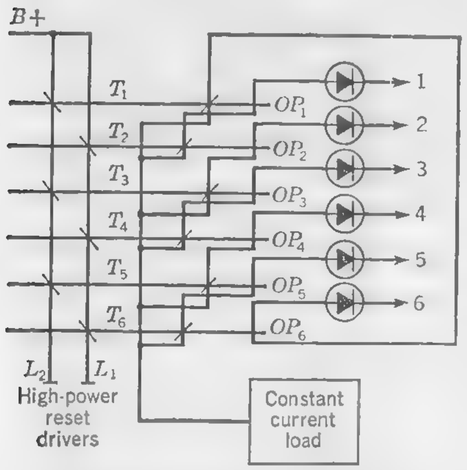


FIG. 12.69. Switch in which output of one core sets the next core in sequence.

In Fig. 12.67*b* the current relationships illustrated are approximately to scale for a typical switch. Considerably more mmf is available at the output of the switch core than is needed either to set or to reset the core. The switch behaves to an extent as a magnetic amplifier and exhibits power gain. This makes it possible to set one switch core from another and still retain enough power to do useful work in the storage arrays. A switch utilizing this principle is shown in Fig. 12.69. Two reset drivers L_1 and L_2 are operated alternately. Suppose core T_1 is set. The reset driver I is operated, and the core produces an output which drives the load and also sets core T_2. The other driver then operates to produce an output from core T_2 and set core T_3. The switch thus acts as a counter and can be used to mechanize complex logic. The power gain in such a system is usually between 6 and 10.

In an actual sequential switch like that in Fig. 12.69, lines L_1 and L_2 do not return directly to $B+$. Instead the drivers are driven from $B-$, and the lines ultimately return to ground. When line L_2 has passed through cores T_5, T_3, and T_1, it then is returned at the bus for output windings OP_1, OP_3, and OP_5. Similarly, line L_1 links cores T_2, T_4, and T_6, and then becomes the bus for output windings OP_2, OP_4, and OP_6. Ultimately these lines return to ground after they have passed through the matrices.

A more conventional diagram of a switch is Fig. 12.70, which shows how the drive lines are brought around to become the busses for the output windings and, also shows a clear winding. The clear winding links core T_6 in the opposite direction to that employed in all the other cores, and it also links T_6 with more mmf than is used for the other cores. In this particular switch, T_6 is considered to be the 0 or initial value of the address which is being defined by the switch.

Figure 12.71 is an actual circuit diagram of a complete 28-way shifting switch which is used to operate a storage unit at a 100-kc rate. The accented line starting at input 1 indicates the path followed by the current when the switch is operated immediately after a clear operation. This current passes down through the primary reset windings of all even-numbered transformers and is then steered by means of switch action through the output circuit of transformer 28. The current continues through the set winding of transformer T_1 and is then routed from that position to X line 1 in the storage matrices. The next drive action will be used to energize input 2. The current in this case will pass through the primary reset winding of all the odd-numbered transformers, will return down the bus, and will be routed by the output winding of transformer T_1 through the set winding of transformer T_2 to X line 2. This action will continue as long as currents are applied to inputs 1 and 2 alternately.

Undirectional Switches—Transistors in the Secondary Circuit. Transistors capable of handling the same peak currents as diodes are now being used for switching purposes. It is possible to modify unidirectional switches so that the base-emitter portion of a transistor replaces the diode. Several advantages result because the output

is taken from the collector circuit. First, the output from the collector is at much higher impedance than is available in the base-emitter circuit. The transistor therefore acts as an impedance amplifier, providing a very high output impedance into the matrices. Secondly, the switch cores are providing only the power necessary to drive the base-emitter circuit and not the power necessary to drive the matrices. Therefore very small cores of storage-core size may be used for the switching operation. This, in turn, increases the number of cores which may be driven by a single set of drivers. Large switches incorporating the base-emitter portions of transistors can be manufactured. Since the output from the transistors is not only at high impedance but can also tolerate very large voltage swings, it is possible for these large switches to drive large matrix arrays. Such switches are useful in the construction of large sequentially addressed storage systems.

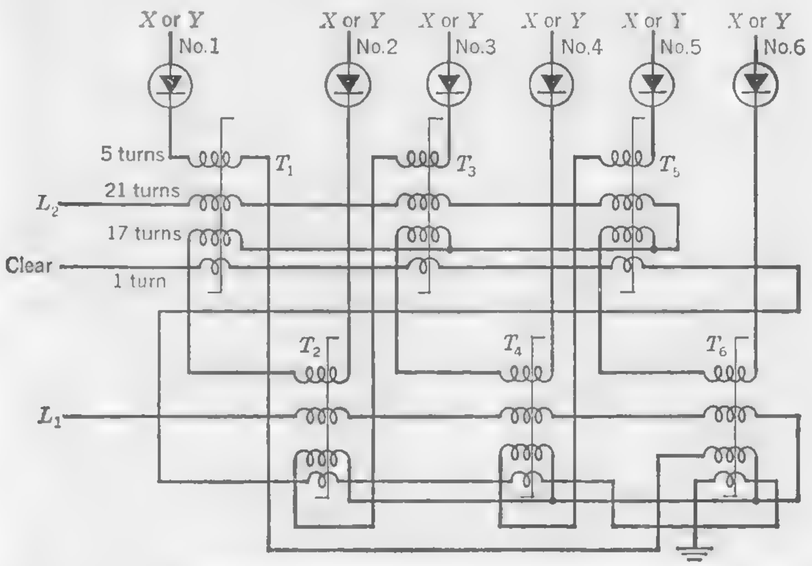


FIG. 12.70. Conventional diagram of a sequential unidirectional switch.

A transistor switch can be arranged to count either in a forward or a reverse direction by energizing the appropriate windings in the switch. Figure 12.72 is a diagram of such a switch. It is used to drive the 120 X lines in a memory system and to scan these X lines in either a forward or a reverse direction as required. The reversible switch steers current to the 120 X lines of the storage array, selecting each line sequentially in either an ascending or descending order. It consists of 121 ferrite cores, each of which controls the operation of a transistor which, in turn, controls the flow of current through one of the 120 X lines. The emitter of the transistor associated with the 121st core is grounded. The function of this extra core is to start the reverse count if 120 words have been loaded. Each core is linked by six windings, five of them with six turns and one with 12 turns. As shown in Fig. 12.72, the five six-turn windings include the clear windings, two drive windings, and the forward and reverse count windings. The 12-turn winding, which is connected between the emitter and base of the transistor, drives the transistor. The windings are arranged to saturate the transistor during the resetting action of the core. At all other times, all transistors are cut off.

The clear winding linking all 121 cores is reverse-wound for core T_0. Core T_0 will be cleared to an initial 1 state while all others are reset to 0. A three-beat drive is used. After clearing, drive 0 will be the first line to be energized. The pulse through this line will reset T_0, driving Q_0 into conductance and allowing current to flow through X 0 and to set T_1. Note that drive 0 does not link T_1 and therefore does not oppose the setting of this core. Drive 1 is the next line to be energized. It will reset T_1, turning on Q_1 and permitting the current to flow through X 1 and both

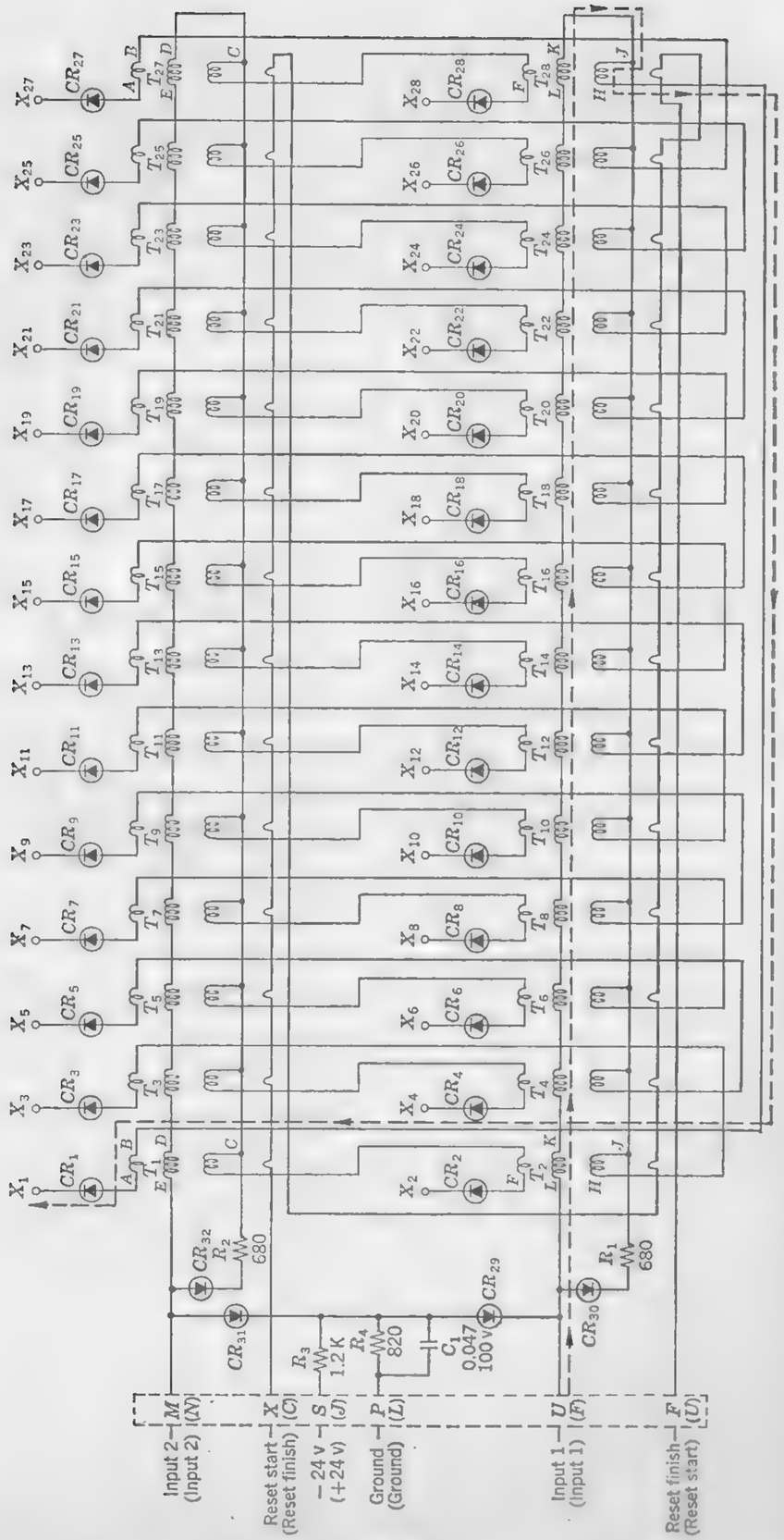


Fig. 12.71. Twenty-eight-way shifting switch.

the "forward" count windings of T_2 and the "reverse" count windings of T_0. Drive 1 will oppose the setting of T_0 but will not oppose the setting of T_2. In this manner, the switch is set sequentially as long as the three drive lines are energized sequentially. The operation is the same for both loading and unloading in the normal cycle. After it ends with the setting of T_{120} or sooner, the next pulse will always be "clear" to restore the initial operating conditions.

To reverse the action, the drive lines should be set in reverse numerical order. For example, if drive line 2 was the last to be energized, reversal will occur if drive line 1 is energized next instead of drive line 0. Once reversal has occurred, the reverse action will continue until a further reversal of drive sequence takes place. The first drive pulse after a reversal should be considered as an extra pulse. It is necessary to

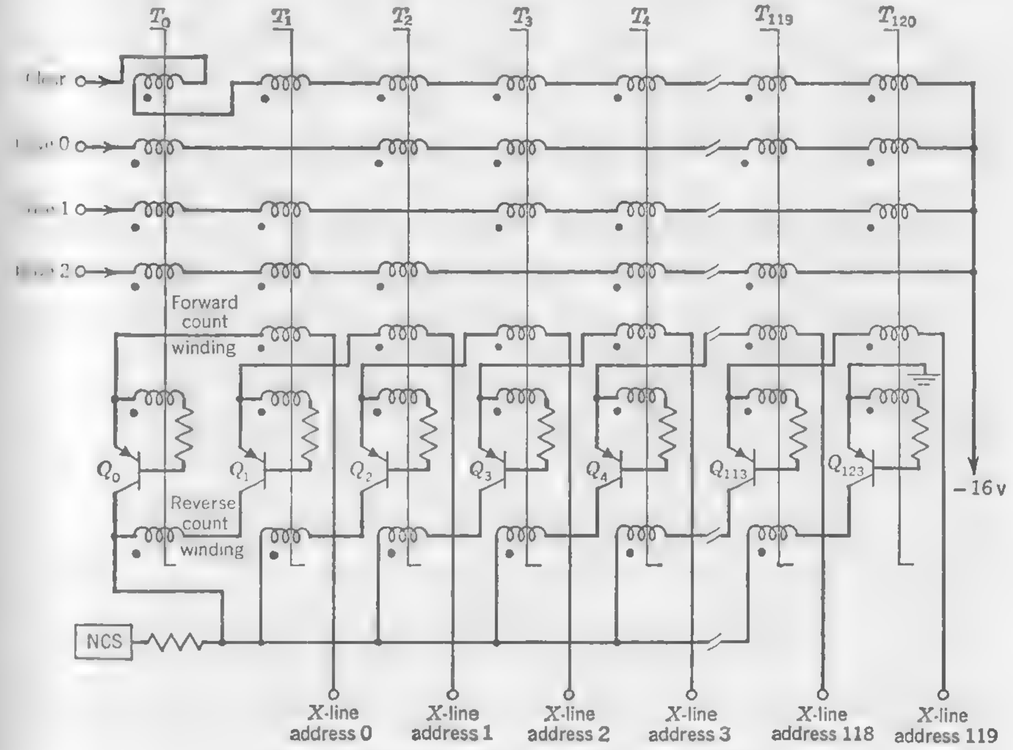


Fig. 12.72. Reversible transistor core switch.

apply this extra pulse to return the core to the last-used position before reversal occurred. For example, if core 3 was the last used, it would have been as a result of energizing drive line 0, and core 4 would also have been set. The first pulse of the reversal process will be produced by driving on line 2, causing core 4 to reset and energizing transistor Q_4. The next winding on drive line 2 (not shown) will prevent core 5 from being set; therefore, the reverse count winding Q_4 will set core T_3, while the action of the forward count winding will be inhibited by the current flowing in drive line 2. It should be noted that, during the reversal action, X line 4 will be energized. The entire system must be designed so that this additional line will become energized in the process of effecting a reversal.

Bidirectional Switches—Diodes in the Primary Circuit. There are considerable advantages to employing diodes in the primary circuits of magnetic switches. First, they can add to the nonlinearity of the switch core or can provide all the nonlinearity in the circuit. Secondly, they can be used to reduce the load presented by the switch to the drivers. This is of considerable importance in large storage units or in very-high-speed storage units. Unidirectional switches with diodes in both the primary and secondary circuits are operated by a combination of the two techniques.

Figure 12.73 shows a typical small switch employing diodes in the primary circuits

of the transformers. In operation, first one of the sinks, then one of the drivers is energized. If sink P and driver A are energized together, current can flow through only one of the primary windings of transformer T_1, and a pulse appears at the output of this transformer in OP_1. Later driver A must be energized, routing current to sink P through the other primary of transformer T_1 to produce an output in OP_1. The two primaries of each transformer are wound in opposite directions so that one can set the core while the other resets the core. If a diode is not included in the secondary circuit, the switch acts like a conventional magnetic-core switch in that it produces biphase outputs at its output terminals.

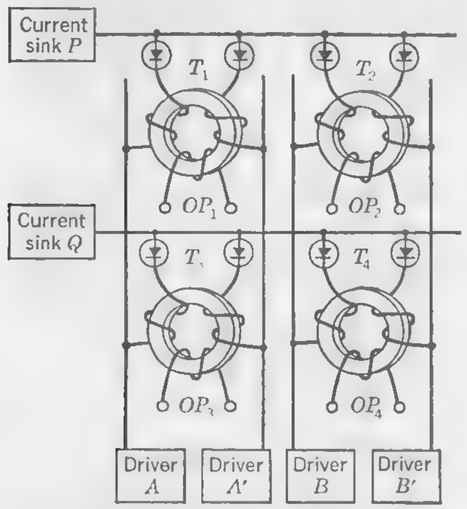


FIG. 12.73. Small switch with diodes in the primary circuits of the transformers.

If the cores in this switch (Fig. 12.73) are made of rectangular-loop material driver A may drive for as long as is necessary and may then be turned off. No further action need proceed until it is desired to produce a reverse output; then driver A' should be energized. Since the current flow through the primary is selected entirely by the diodes, with a switch of this type it is possible to separate the load and unload actions in the cycle. For example, if a drive such as AP is used to produce the read action, then a drive such as $A'P$ is used to produce a write action. If a switch is used in this manner, the operating technique must be such that an $A'P$ action always occurs before a second AP action in an individual core.

Another way of mechanizing the switch is to replace all the A', B', etc., drivers and their associated windings with a single constant-current winding threading all switch cores. This winding must be capable of resetting any of the switches. The drive handled by the remaining drivers and the current sinks must then be doubled in previous value. A read action is initiated by a connection such as AP; but when driver A is no longer energized, a reverse output is obtained immediately from core P under the action of the common bias winding. This type of switch has considerable advantages. It requires a small number of drivers, and its cycle time is short since a write action automatically follows a read action with maximum speed. Its disadvantages are the large size of the drivers and the necessity of controlling the trailing edge of the current from the drivers. This edge forms the rising edge of the write pulse from the output windings.

The cores in the switch need not be of square-loop materials. In fact, cores of linear material have lower coercive forces than square-loop cores. If linear-type cores are used, they act as transformers and the switching action is controlled entirely by the diodes. Extremely rapid operation is obtained with these switch cores, making them suitable for high-speed storage units.

The Secondary Circuit of a Switch Core. The load presented by a matrix has linear elements of resistance, capacitance, and inductance, as well as a nonlinear element which contributes varying capacitance, inductance, and resistance values. The pure inductance is by far the greatest quantity, with the nonlinear inductance, capacitance, and resistance following in that order. During switching, the switch core behaves as a relatively high impedance and sets up a current through the load. When the core is saturated, this current continues to flow because the secondary winding then becomes a small, relatively linear inductance. The natural time constant under these circumstances is very long; and usually it is necessary to add resistance in series with the load to shorten the saturated secondary-circuit time constant. Additional resistance can be provided by resistors at the end of the winding or can be distributed by using a resistive wire for the matrix windings.

The instantaneous voltage across the load during the operation of the switch

depends on the number of storage cores which are moved from one state to another by coincident-current action. For example, in a system driving a 40-bit parallel word any number from 0 to 40 cores can turn over. In the 1-μsec materials this can represent a change of 4 volts. The entire secondary voltage may be no higher than 15 volts and would probably be no higher than 40 volts. This represents a change of 10 to 25 per cent in the instantaneous value of the load. If all the cores turn over, the switch core must provide more flux to the load than if none of them turns over. During a reading cycle the same amount of flux is required during both the set and the reset actions. During a write cycle, however, a word of all 0's may be changed to a word of all 1's, and there is a considerable flux variation between set and reset. Ordinarily no flux can be transferred through a transformer, and the flux absorbed by the load must be exactly equal to the flux change in the transformer.

In the above-mentioned example, a small amount of flux is required during the setting action, and only this amount is available on reset. Unless special measures are taken, it would be impossible to alter the state of the storage cores. One such measure is to add to the load a resistance which acts as a flux-dissipating device. The resistance allows the transformer to move through a relatively fixed flux change while the duration of the secondary reset pulse varies slightly to compensate for load variations. The correct value of resistance must be found by experiment. Generally it is higher than the value required for correct damping in the secondary. In order to obtain reliable operation, the voltage drop in the resistance should be about ten times the amount by which the load voltage can change with alteration of information content. Some storage units have been operated successfully when this figure is as low as 5, but unless particular care is taken in the design to control the operating currents, marginal operation can be expected.

12.7.14. Diode and Transistor Switches. Figure 12.74 shows a typical arrangement for diode decoding. The diagram indicates current flow in one direction only in the matrix lines and in one set of matrix lines only. In a typical system there

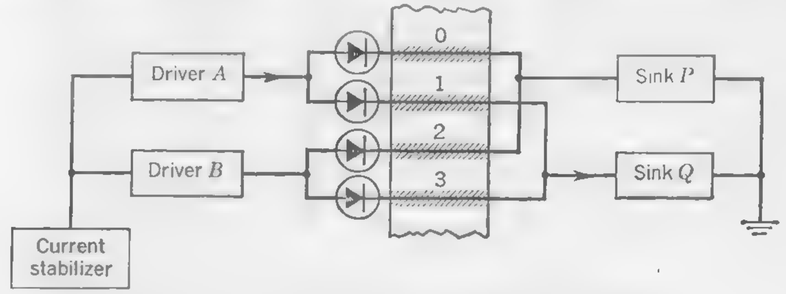


FIG. 12.74. Diode decoding switching arrangement.

will be 64 such diodes, eight drivers, and eight sinks; and in a complete 4,096-word memory system, there will be four sets of drivers, diodes, and sinks, or a total of 32 drivers, 32 sinks, and 256 diodes. Superficially, diode decoding switching does not appear to be very efficient since a large number of components is employed and the combinatorial advantage is no greater than can be achieved with magnetic switches. However, the impedance match between transistors and cores is so good that a medium-capacity memory can be driven efficiently from a single set of transistor drivers and sinks. If a magnetic-core switch is interposed between the drivers and the matrix, the loss introduced by the core switches is usually sufficient to necessitate increasing the number of transistors in the drivers and sinks.

The technique of decoding with diodes directly into matrices can frequently introduce a considerable cost reduction in large systems. Where very-high-speed switching is required, this is one of the few feasible methods. To date no magnetic switch has been designed to operate at the high speed necessary to drive the fastest available storage cores. In linear- or word-selection memories the complete selection is performed by diode switching. In a 1,024-word memory, for example, there would be 32 drivers, 32 sinks, and 1,024 diodes for each direction of current flow in the matrix. Low-speed linear-selection storage devices can be built with magnetic-core switches,

but they are usually less economical than coincident-current devices operated by core or diode switching.

Operation of the diode decoding scheme in Fig. 12.74 is as follows: If driver A and sink Q are selected, the current flows from the current stabilizer through driver A and through the diodes on lines 0 and 1. These lines behave as short-constant delay networks, and initially the currents divide equally down lines 0 and 1. However, the current flowing in line 0 encounters an open circuit at the end of that line, with reference to sink P, and flows up line 2 toward the diode. The current in line 1 encounters a short circuit because sink Q is conducting; it therefore flows into sink Q. Meanwhile the current which is flowing in line 2 finally encounters an open circuit at the diode on line 2 and is reflected back down line 2 and then up line 0 to establish an open-circuit condition at the input end of line 0. Four complete propagation times through the matrix are required to establish the correct current in line 1. In practice, the lines do not encounter perfect open-circuit or short-circuit terminations at their ends. A total of six to eight propagation times should be allowed to establish completely the current in the selected lines. An alternate method is to bias the matrices so that an open circuit is encountered at the input end of every line. Then if sink Q is energized in advance of driver A or B, only the diode on line 1 will conduct when driver A conducts, and the current conditions are quickly established with respect to driver A.

Transistor Switches. It is also feasible to switch directly into matrices by transistors instead of diodes. This has already been discussed to some extent in the description of a transistor core switch. A typical transistor switching scheme is shown in Fig. 12.75. The transformers shown can be elements of a switch or can be linear transformers driven from low-level decoding logic ahead of the drivers. The positive and negative current stabilizers feed a large number of these transistor switches. The advantage of driving by transistors is that large voltage swings can be tolerated at the output circuit. Transistor driving is therefore suitable for large memory systems, particularly when high-speed operation is desired. Transistors may also be used as power amplifiers in a large magnetic switch with low output power. By this means, it is possible to obtain the combinatorial advantages of a magnetic switch with the impedance-matching advantages of transistors.

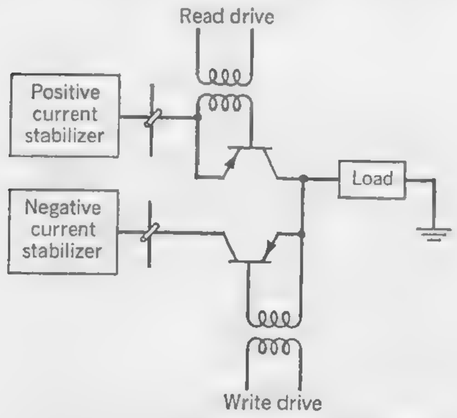


FIG. 12.75. Transistor switching scheme.

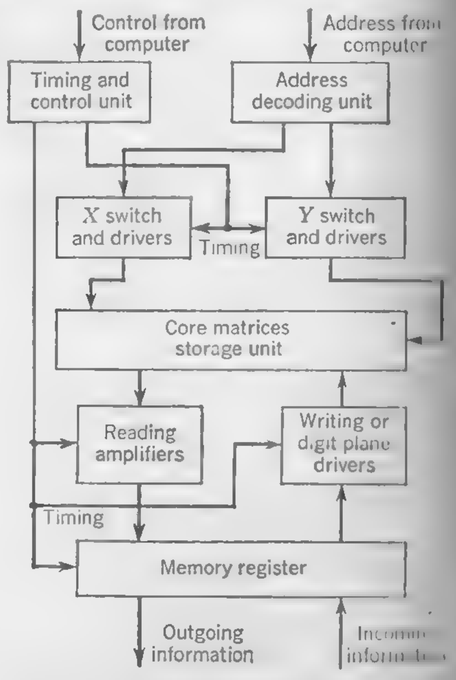


FIG. 12.76. Block schematic of a coincident-current storage unit.

Storage Units and System Design

12.7.15. Coincident-current Storage Units. Figure 12.76 shows a general block schematic of a coincident-current storage unit. The units at the top of the diagram perform the necessary timing and control and produce partially decoded address waveforms. These units employ conventional computing-type circuits which provide the X and Y drives to the matrix storage unit. Some of the blocks, such as the memory register, are omitted in certain storage systems, but they will be found to exist in some form in the computer.

There is at least one reading amplifier for each parallel bit stored, and in very large units several amplifiers per bit are used to reduce the length of each sense winding. The reading amplifier, a suitable push-pull stable amplifier, must offer high rejection to in-phase signals. It must have good low-pass characteristics and must not be paralyzed by noise spikes which are somewhat larger than the amplitude of the desired signal. The final component is a gated discriminator which emits either 0 or 1 signals under the control of a strobe pulse. If the delay in propagation through the matrices is large, several strobe pulses are employed so that each reading waveform is strobed at the optimum time.

The parallel output from the reading amplifiers sets the memory register which, in turn, transmits the information to the computer. The memory register, which controls the rewrite action during a read cycle, is loaded from the computer to control the write action. The writing or digit-plane drivers are precise pulsed current drivers which cause the information to be inserted into the matrices.

In serial machines it is common practice to employ a parallel storage unit, which is often more economical than an all-serial unit. Serial-to-parallel and parallel-to-serial converters are inserted in the incoming and outgoing lines. In some serial machines two storage units are used, each storing one-half of the word. These units are then employed in an interlaced fashion so that one unit is writing or rewriting the last half of the word at the last address, while the other unit is reading the first half of the word at the present address. This arrangement leads to rapid serial operation.

12.7.16. Methods of Driving Coincident-current Storage Units. Progress in the design of coincident-current magnetic-core storage systems has always depended on the availability of new storage materials and new tubes or transistors for driving purposes. The investigation of transistor drive had scarcely been started in 1955, but by the end of 1956 every storage unit which was in design used transistor drive entirely or for the major portion of the unit. This rapid transition from tube to transistor drive was due to the superiority of transistors in matching magnetic-core impedances.

Direct Drive. All the original magnetic-core storage systems used direct vacuum-tube drive because switch and storage materials were so poor at the time that they could not be combined into an operating system. A schematic of part of the driver system in an early storage device is shown in Fig. 12.77. It is still typical of circuits that are frequently used in evaluating the performance of matrices. A direct-driven circuit, such as that shown in Fig. 12.75, is used nowadays with transistors replacing the tube drivers in Fig. 12.77 and with only one wire in each direction in the matrix. In the system shown in Fig. 12.77, two X and two Y wires are used for each core in order to utilize the high impedance from the plate circuit for driving into the magnetics. Later systems used a single X or Y wire which was driven by one driver plate and one driver cathode. One matrix has been shown in the diagram, but in practice a number of matrices are usually employed in cascade. In the example shown, the impedance match is very poor because the tubes are capable of delivering much higher voltages than are ever required by the matrices.

Consider a storage unit which is designed to store 4,096 words of 40 bits each. Assume that the system is driven by 64 drivers in each of the X and Y directions. Also assume that the current flowing in the X or Y wires is 200 ma and that the voltage induced in a selected core is about 30 mv and the voltage induced in an unselected core is about 3 mv. It is difficult to determine the exact value for the cores which are energized with only a half current. The first time that a core is disturbed, a large disturb signal will be produced. The amplitude of the disturb signal will decrease with each subsequent disturbance until it reaches a minimum value. In designing a core-storage unit, it is not correct to use either the maximum or minimum value in estimating the average load on the lines. The full potential operating range of the equipment should be examined in order to choose a correct

average value. The number of cores in operation on the X and Y lines is symmetrical. The selected X or Y wire links 63 half-selected cores in each matrix and one selected core. The selected core may or may not switch, depending on whether it is storing a 0, or a 1. For the complete system, each X or Y wire links 2,520 half-selected cores and 40 selected cores. The minimum voltage drop due to the cores themselves, if the word that is read contains all 0's, will be 7.68 volts. To this must be added the voltage drop due to the resistance of the wire and to the two inductances of the windings. Generally, the sum of these two factors is about equal to the voltage drop introduced by the cores. Therefore, it will be assumed that the normal voltage drop through the matrices, if none of the selected cores switches, is a little over 15 volts, say 15.25 volts. If all the selected cores switch, an additional 1.25 volts will be required. In designing the system it should be assumed that the minimum instantaneous voltage drop is about 15 volts and the maximum is about 16½ volts. This

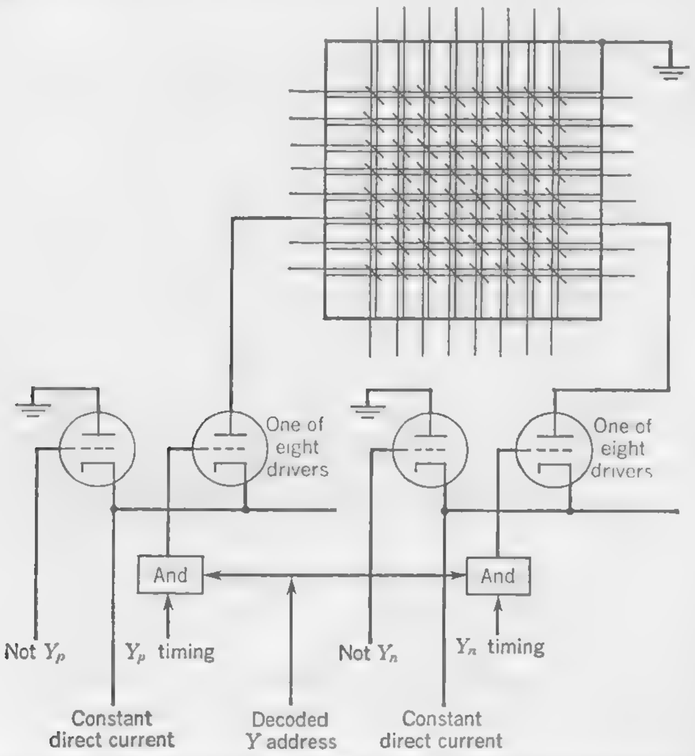


FIG. 12.77. Part of the driver system of an early storage device.

voltage at a current of 200 ma can be delivered by transistors, although the value of the voltage is perhaps a little high for reliable operation. It does indicate that the match between the transistors and the cores is extremely good. In a tube-driven system this voltage would represent an extremely poor match.

Transformer Drive. Linear transformers may be inserted between the drivers and the cores to effect an impedance match. It then becomes necessary either to add resistance in series with the matrix wires or to use resistance wires for windings the matrices. (The use of this resistance was described under the paragraph headed The Secondary Circuit of a Switch Core.) The amount of resistance required in the secondary circuit of a linear transformer is not so great as that needed in a switch core, but some resistance is required to compensate for the varying flux absorbed by the matrices as the information content changes. In addition, since the transformer itself introduces loss and inductance, about twice as much peak power must be supplied to the primary of the transformer as is supplied in a direct-driven system. For this reason, transformer drive is chiefly of value in small-capacity storage units.

Transformer drive is frequently used in conjunction with diodes to form a diode transformer switch. Such a switch effects an impedance match and introduces some combinatorial advantage, which reduces the required number of drivers. Transformer coupling between the drivers and the storage matrices has been used successfully in transistor-driven equipment. In very large systems, transformer coupling is used to step down a current into the matrices when the transistors cannot develop enough voltage to drive the storage matrices. In small systems, it is used to step up the current to the storage matrices. The matrices require more current than can be delivered by the transistor drivers.

Switch Drive. Direct- and transformer-driven systems are simple to design and are particularly desirable for memory units which will not be reproduced in large quantities. If quantity production is to be considered, savings in equipment and cost can be effected by using switch-driven systems. Although they are more difficult to design and test than direct-driven systems, the switch-driven systems are more economical to produce and usually more reliable. Most magnetic-core storage units in wide use today employ a switch of some type in the drive circuits. If a unit is designed for small-quantity production, usually a rather simple switch is employed. Only in high-quantity-production devices are the more complicated types of switch used.

In random-access applications, the choice of a simple type of switch is dictated not only by the economical advantages in design cost and over-all system operation but also by the fact that complex switches are rather inefficient. A typical simple switch-driven system using an anticoincident switch is shown in Fig. 12.78. The

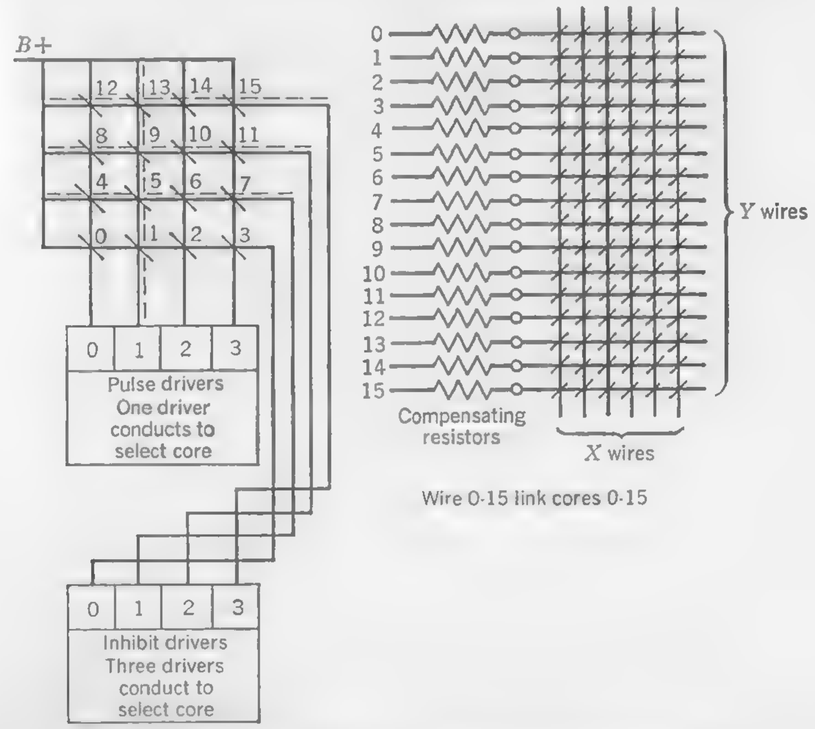

B+
12 13 14 15
8 9 10 11
4 5 6 7
0 1 2 3
0 1 2 3
Pulse drivers
One driver conducts to select core
0 1 2 3
Inhibit drivers
Three drivers conduct to select core
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Y wires
Compensating resistors
X wires
Wire 0-15 link cores 0-15

FIG. 12.78. System driven by an anticoincident switch.

broken lines in the switch matrix indicate the wires that are energized. In this instance, line 1 in the vertical direction and lines 1, 2, and 3 in the horizontal direction are energized. The current in the horizontal direction inhibits the switch, while the current in the vertical direction drives the switch. Only core 1 is driven and not inhibited; therefore, only this core produces an output. The connections between

the output circuits of the switch cores 0 through 15 and the compensating resistors numbered 0 through 15 are not shown in order to simplify the diagram.

There are many kinds of double-coincident switch in use today. They all have similar properties and utilize about the same number of drivers for a given size of switch. The choice of one form of switch or another is usually made on economic rather than on technical grounds. The cost savings introduced by a specific switch cannot be stated in any general form. Various types of switches must be sketched out in a fairly complete design to decide which is the most economical for a particular system. Generally, a switch-driven system will use only about 20 per cent of the components in the driver circuits that would be required in a direct-driven system. On a similar scale, a transformer-driven system will use somewhere between 50 and 60 per cent of the components that would be used in a direct-driven system. These figures can be misleading because the switch may be a costly item to manufacture.

Mixed systems in which one axis of the storage planes is switch-driven and the other axis is direct-driven or transformer-driven are occasionally fabricated. They are particularly effective when it is necessary to obtain accurate control of the current in one direction, usually in the axis which is direct- or transformer-driven.

Special-purpose Switch Systems. Sometimes storage units are required which refer to each address in a sequential manner rather than in a selective manner. A unit of this type permits the designer to achieve an increase in circuit economy. The switches illustrated in Figs. 12.67 through 12.71, 12.74, and 12.75 are useful in this respect, because they act both as counters and as switches. They remove the necessity for address circuits and, in addition, require very few drivers to operate them. Two switches of this type may be used in the X and Y directions as long as the number of outputs from one switch is mutually prime with respect to the number of outputs from the other switch. This requirement can be mechanized in a simple fashion if one of the switches has three drivers instead of the two illustrated in Fig. 12.67. Both switches will count together, and ultimately every core in the matrix will be scanned as indicated in Table 12.28.

Table 12.28. Scanning Pattern in a Special-purpose Switch System

X switch driver	X switch position	Y switch driver	Y switch position	Core which is scanned
1	1	1	1	1
2	2	2	2	6
1	3	3	3	11
2	4	1	1	4
1	1	2	2	5
2	2	3	3	10
1	3	1	1	3
2	4	2	2	8
1	1	3	3	9
2	2	1	1	2
1	3	2	2	7
2	4	3	3	12

In most storage systems of this type, it is not important to retain the information once it has been read. Then it is convenient to employ two sets of switches, one of which operates during a write operation and the other during read. Because each switch remembers the address of the next operation, it is possible to write in a block of information, read out some of it, write in some more, and so on. In this type of operation only one beat is required either to write or to read since every reading operation clears the storage cell to 0 in readiness for a write operation.

The information may be regenerated, and the operation of the storage unit may be speeded up, by employing a special matrix configuration in conjunction with the

shifting switches. A system of this type is illustrated in Fig. 12.79. The matrix wires are taken through a row or column of cores and are then returned in the opposite polarity through the previous row or column. Wires in the first row or column are taken back through the last row or column. If, for example, core 6 is being set to the 0 state, or read, then core 1 which was the last core set to 0 is in process of being set into the 1 state. Two reading amplifiers and two digit-plane drivers and their

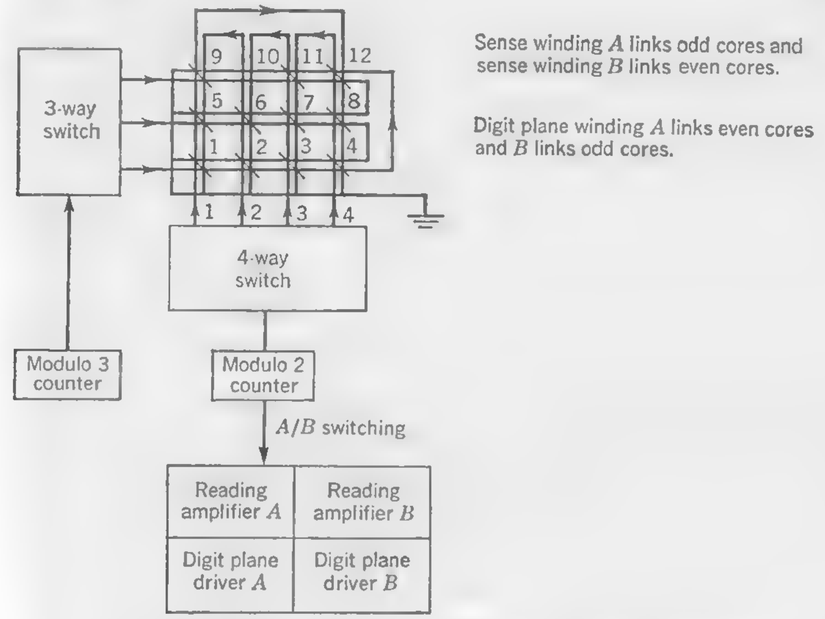


FIG. 12.79. Special-purpose switch system for sequential operation.

associated windings are used alternately. The operating sequence is shown in Table 12.29. The system reads and writes simultaneously and therefore operates twice as fast as a conventional selective system. It is economical with respect to components, but the complexity of the matrix windings limits its application to small storage units.

Table 12.29. Operating Sequence in a Special-purpose Switch System

4-way switch output	Reading amplifier in use	Digit-plane driver in use	Core read	Core write
Odd.....	A	A	1	12
Even.....	B	B	6	1
Odd.....	A	A	11	6
Even.....	B	B	4	11 and so on

The diode-decoding scheme illustrated in Fig. 12.74 is also used to achieve additional decoding associated with the storage unit itself. Diode decoding may be used in conjunction with a direct-driven system or to place two magnetic switches in cascade. Essentially the system is arranged so that one switch produces a positive voltage while the other produces a negative voltage. The selected diode is the only one that can conduct as a result of the positive and negative operations of the two switches in cascade. In order to operate a system in this manner, there must be close control of the d-c bias which is used as part of the selection operation and of the outputs of the switches. To achieve adequate control of the voltages, switches employed

in cascade are usually unidirectional steering switches in which the prime current is drawn from a current stabilizer of some type.

Diode decoding in conjunction with magnetic switches has been used successfully in sequentially accessed linear- or word-selection memories. In a memory of this type the current through the diode is allowed to charge a shunt inductance at the same time as a large pulse of current is routed down the matrix wire to perform the read operation. When the drive switch has ceased to operate, the current flowing in this inductance causes a current reversal in the matrix wires which is low in amplitude and long in duration compared with the original current pulse applied by the switches. Such a current-generating technique is adaptable in either a random-access or a sequential-access unit in which only one set of unidirectional switches is used to initiate the read operation. The write operation follows automatically as a result of the overswing of current due to the inductances.

12.7.17. Circuits Peculiar to Core-storage Units. In most core-storage units, standard circuits are used for flip-flops, gates, delay elements, timing elements, and for all the usual logical functions. In those units designed to achieve high reliability special types of core logic replace the conventional transistor-diode logical circuits. There are two types of circuit which are truly peculiar to the magnetic-core storage unit. The first are the driver circuits which insert addresses and information into the memory, and the second are the reading or sense circuits which detect the output from the magnetic-core array.

The electronic circuits associated with a storage unit must be extremely reliable because errors introduced in the memory are a severe handicap to the user of digital equipment. All circuits, whether or not associated with the memory, are toleranced carefully to achieve a high probability of long life. Components are derated and toleranced to operate at a much higher temperature than would be expected within the equipment. They are rigorously inspected and tested before assembly, and a sample number are tested to destruction. Power supplies are regulated to reduce the effects of line variation, and circuits are toleranced to withstand large variations in power.

Circuits described below are typical of those in high-quality equipment.

Current Stabilizer. The current stabilizer is an important element in many drive circuits. It delivers a constant regulated current at a high impedance which may be steered by means of transistor switches into the magnetic portion of a storage unit. A typical current stabilizer is shown in Fig. 12.80. This circuit is designed to pro-

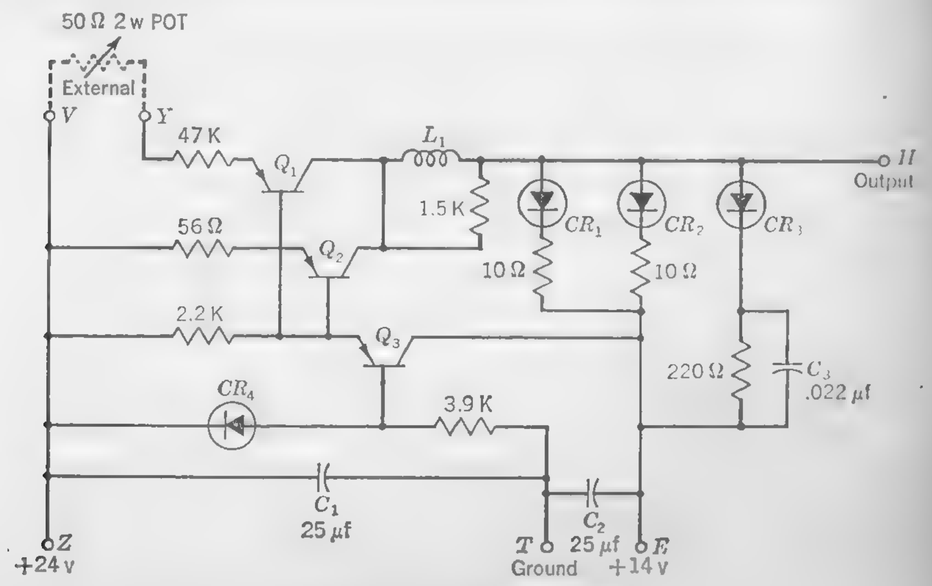


Fig. 12.80. Current stabilizer.

duce a constant current regardless of the operating temperature. Other circuits are temperature-compensated to vary their current output and compensate for the changing characteristics of the switch core effected by variations in temperature. This circuit maintains current at a constant value regardless of load-voltage excursions ranging from 0 volt (ground) to close to +14 volts. The magnitude of the current may be adjusted with an externally mounted potentiometer to 185 ma. The diode CR_4 acts as the regulating element and may be considered a constant battery of +6 volts placed across the emitter-base section of Q_1 and Q_2. The L_1 in the collector circuit is a surge-limiting element.

Transistor Switch. Transistor-switch circuits are used in routing current from a current stabilizer to a magnetic portion of a storage unit. A typical circuit is illustrated in Fig. 12.81. Transistor Q_3 serves as a unidirectional switch with both its terminals floating so that the switch may be connected between a variety of power sources and loads within capabilities of the circuit. The power source may be of either polarity and of either the constant-voltage or the constant-current type.

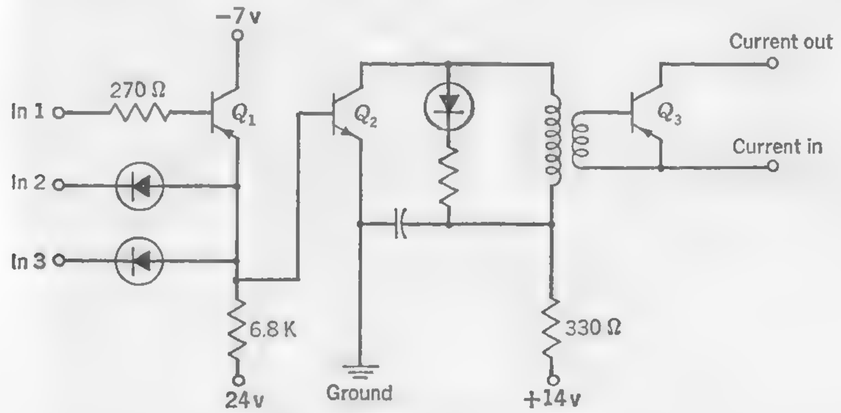


Fig. 12.81. Transistor switch circuit.

The versatility of the circuit is due to transformer coupling of the enabling pulse to the base of Q_3. The transformer is also the source of two limitations of the circuit: (1) the maximum duration of switch closure is 10 μsec, and (2) the recovery time is 10 μsec.

Transistor Q_1 and the diodes at the inputs to the circuit form an AND gate whose nominal operating voltage levels are ±5 volts. Transistor Q_2 serves as an amplifier between the gate and the switch Q_3. If any one (or more) of the input levels is 5 volts, the switch will be open. When all input levels are +5 volts, the output of the gate will rise and cause Q_2 to conduct, which in turn energizes Q_3 to conduction.

Because of its time limitations, the switch must be turned off within 10 μsec and must not be reenergized for another 10 μsec. This timing is usually accomplished with pulses of a suitable duration and a duty cycle applied to the base of Q_1. The maximum current permitted to flow through a closed switch is 200 ma. The rise time of current to 200 ma through a resistor load is not more than 1 μsec.

The above circuit is used in small storage units. Transistor switch circuits used in large storage units operate in a similar manner but are designed to handle much larger currents and voltages.

Digit Driver. Digit-driver circuits are used to insert information into the storage matrices. They are also used for inhibit purposes when driving switches and as sinks when driving diode core switches or in applications requiring diode decoding in association with storage matrices.

A typical digit-driver circuit, shown in Fig. 12.82, functions as a switch. One of its two terminals is connected to ground; the other (output) may be connected to a negative supply through a suitable load. The switch is closed when both the information input and the $\overline{IU}$ input are at −5 volt, and open when either input is at

+5 volts. The specific function of the circuit is to hold a selected core in the 0 state by energizing the inhibit winding passing through the matrix cores whenever the information input is a 0 (−5 volt).

The circuit has an input stage, consisting of transistors Q_1 and Q_2 plus four clamping diodes; a difference amplifier, Q_3 and Q_4; and a transistor switch, Q_5 and Q_6, that forms the output stage. The information input is clamped to ±5 volts; the $\overline{LU}$ input is at +5 volts during UNLOAD and −5 volt during LOAD. The emitter followers, Q_1 and Q_2, form a negative AND gate for these inputs. When both inputs are at −5 volt, Q_1 and Q_2 will conduct very little and drive the transistor switch out of conduction through the emitter follower Q_3.

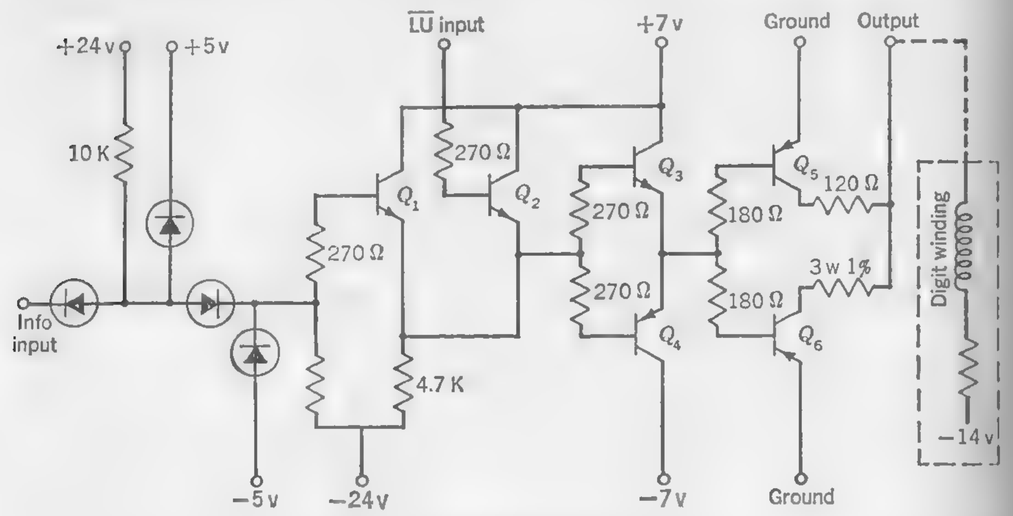


FIG. 12.82. Digit-driver circuit.

The output circuit consists of current-limiting resistances in series with a digit winding and the switch. It is energized by a −14-volt supply. A maximum of 15 volts is permitted across the switch when it is open, and a maximum current of 200 ma is permitted to flow through a closed switch. The rise and fall times of the currents through a resistive load are not more than 1 μsec.

The digit-driver circuit described above is designed for applications where the current is kept constant with temperature. If it is desired to compensate for temperature change, instead of using −14 volt in the collector circuit of Q_5 and Q_6 a special power supply is used in which the change of voltage with temperature is proportional to the change of current that is required in the cores.

Read Amplifier. The output signal from the sense winding of the core matrix consists of two types of signal. The first is a large in-phase excursion of the sense winding due to electrostatic coupling between this winding and the X and Y wires in the matrix. Both ends of the sense winding go through this excursion together if the sense amplifier and sense winding are correctly designed. A second signal, between the two output wires, is the desired read signal. The ratio between the undesired electrostatically induced in-phase signal and the desired magnetically induced out-of-phase signal is never less than 100:1 and in large memories is frequently as high as 1,000:1. In order to detect the desired signal, the read amplifier must be a well balanced and well-designed difference amplifier.

A typical read-amplifier circuit used in very small storage devices is shown in Fig. 12.83. This circuit accepts bidirectional turnover signals from a magnetic matrix core. A signal is first amplified and then sampled at its peak amplitude with a strobe pulse. The circuit consists of an input pulse transformer, a highly sensitive input amplifier, and a transistor gate. Both ends of the matrix sense winding are connected to the primary of the input transformer. Depending on its polarity, the

input signal is amplified by either Q_1 or Q_3 and is applied to the base of the transistor gate Q_2. The strobe pulse is inserted into the emitter of the transistor gate Q_2.

To provide proper detection, the turnover signal from the sense winding must have a peak amplitude of 20 mv or more, and the duration of the signal at 5 mv must be 3 μsec or more. Following detection of such a signal, the output pulse will be a replica of the strobe, rising from its normal level of −5 to +5 volts in not more than 1 μsec, remaining at this level for not less than 1 μsec, and falling to the original −5-volt level in less than 1 μsec. It remains at the −5-volt level when no turnover signal is detected.

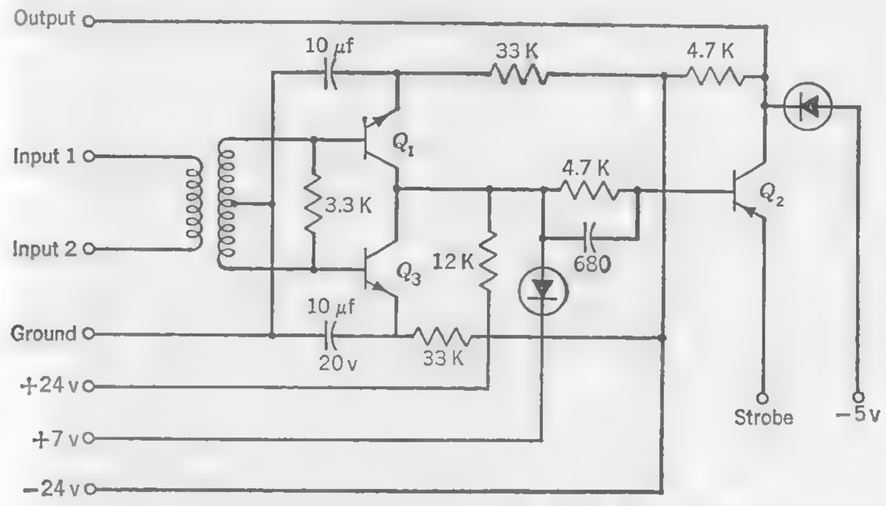


FIG. 12.83. Read-amplifier circuit for small storage unit.

A more sophisticated read-amplifier circuit which is used in large storage units is shown in Fig. 12.84. The input signal is applied to the primary of transformer T_1. The secondary windings of T_1 provide a balanced push-pull input to transistors Q_1 and Q_2. The positive-going edge of the pulse input provides a voltage at the base of Q_1 that is positive with respect to the voltage at the base of Q_2. Under these conditions, Q_1 reduces conduction and Q_2 increases conduction. The collector output voltage of Q_1 drops and the collector output voltage of Q_2 rises, resulting in a negative voltage at the base of transistor Q_3 with respect to the voltage applied to the base of transistor Q_4. Transistors Q_3 and Q_4 comprise a differential amplifier circuit with cross coupling accomplished by capacitors C_1 and C_2. The resultant increase in conduction through Q_3 provides a positive voltage at the junction of resistors R_{13} and R_{15}. This rise in voltage is fed back to the emitter of Q_1 through capacitors C_3 and C_4 to stabilize operation. The positive voltage at the collector of Q_3 is also applied through capacitor C_9 to the junction of diodes CR_1 and CR_3. Decreased conduction through Q_4 produces a negative voltage at the junction of resistors R_{14} and R_{16}. This drop in voltage is fed back through capacitors C_5 and C_6 to the emitter of Q_2 to stabilize operation. The negative voltage at the collector of Q_4 is also applied through capacitor C_{10} to the junction of diodes CR_2 and CR_4. Capacitors C_7 and C_8 in the collector circuits of Q_3 and Q_4 maintain a sharp pulse edge on the collector feedback voltage. The voltage at the junction of CR_1 and CR_3 is positive with respect to the voltage at the junction of CR_2 and CR_4. Resultant conduction through the diodes produces a negative voltage at the junction of CR_1 and CR_2 with respect to the voltage at the junction of CR_3 and CR_4. The voltage at the junction of CR_1 and CR_2 is directly applied to the base of transistor Q_5.

An output is obtained from the read amplifier only when the read strobe, generated in the timing and control circuits, is applied in coincidence with the input from the associated read winding in the memory module. The normal bias level at the base of Q_3 is +6 volts, established by the divider R_{18} and R_{19}. The read strobe is also

referenced to +6 volts, so that the voltage at the Q_6 base biases Q_6 1 volt into cutoff. Only when the read strobe and the signal go negative together by more than 1 volt does Q_6 conduct, establishing +5 volts at the emitter of Q_7. If either the read strobe or the rectifier remains above +5 volts, Q_5 remains cut off. The base of Q_7 is clamped to −5 volt by CR_6.

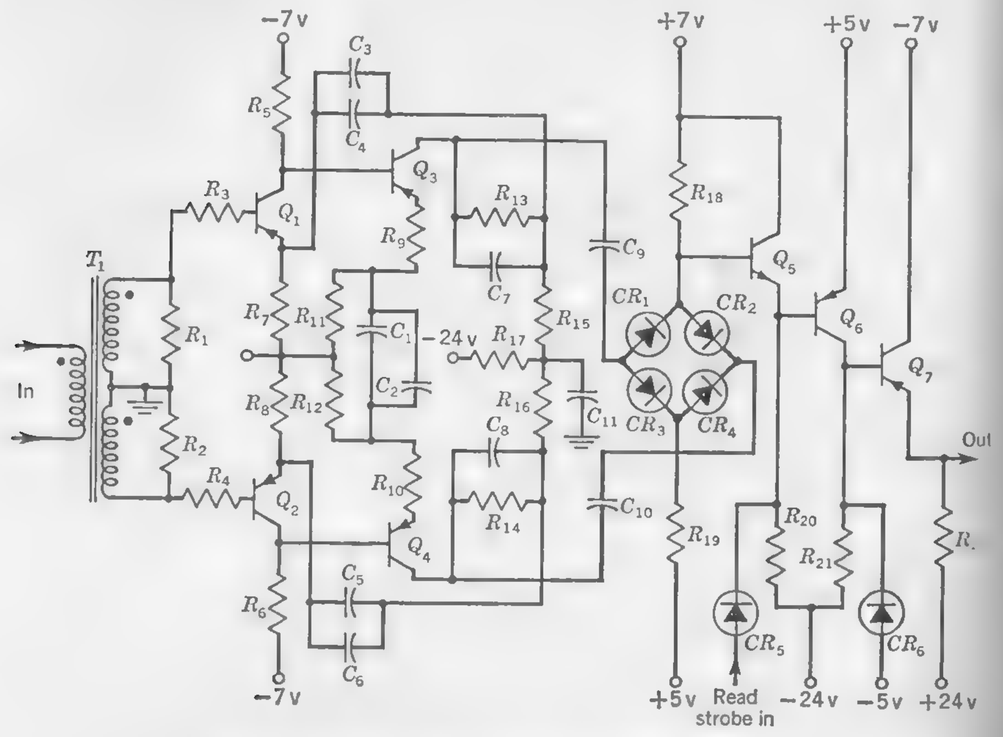


FIG. 12.84. Read-amplifier circuit for large storage unit.

12.7.18. Typical Magnetic-core Storage Units. Magnetic-core storage units have evolved into two major classes. The first class are the general-purpose units which are random-access in nature and of relatively large capacity. Capacities range from as low as 10,000 binary digits to more than 1 million binary digits. Units of this class are characterized by their random-access properties and by the utilization of a two-beat cycle in which both reading and writing occur. In most of them it is possible to perform either a read-and-regenerate cycle or a clear-and-write cycle. In some of them the cycle is split so that independent synchronizing pulses may be used to initiate the read and write operations. During the first part of the cycle, a reading operation always takes place, and in this process the selected word is cleared to all 0's. The second part of the cycle must always occur but may be delayed by an indefinite period before it occurs. During the second part of the cycle, the information that was read is regenerated or a new word is written into the selected memory cell. Every cell of the memory is effectively an accumulator since a word can be brought down from the memory, modified, and inserted back into the memory in the last part of the cycle.

The rate of operation of the general-purpose storage units varies from 1 μsec per cycle in the high-speed units to low speeds of 20 to 50 μsec per cycle. The cycle time is defined as the interval from initiating a complete action in the memory until the next complete action can be initiated. In random-access memories, the cycle time includes a complete read-and-regenerate operation or a complete clear-and-write operation as well as the time necessary for setting up a new address between cycles. The access time of the storage unit is defined as the delay between the moment when all information is fed to the memory to instruct it to perform an oper-

tion and the earliest moment at which the information stored at a particular address is available in an output register. This time varies from about ½ to about 10 μsec.

The second major class of core-storage units are special-purpose units, also referred to as buffer storage units. Early units of this type were buffers designed to effect time matching between two devices operating at different speeds. They were found to be valuable in changing the format of the information during the time-matching operation. Special-purpose storage devices perform pure storage functions as well as a number of logical functions usually associated with the computing art rather than with the storage art. These storage units are usually small in capacity, rarely exceeding 10,000 binary digits. The minimum capacity of a buffer storage unit may be as low as 100 or 200 bits. Below this size, it is usually more economical to use magnetic-core shift registers.

The original type of buffer storage unit was designed to transfer information from one continuous medium to another, such as paper tape to magnetic tape or magnetic tape of one type to another type of magnetic tape. It stores information as characters of 4 to 8 bits parallel rather than as words. Its capacity is small, usually between a hundred and a few thousand characters. It contains two sets of counters, one of which remembers the load address and the other the unload address. The order of the information passing through the buffer storage unit is unaltered. Loading and unloading operations can be intermixed in any way desired. The mode of operation of the unit is illustrated in Fig. 12.85. At the beginning, the unit is cleared and both the load and unload addresses are set to the initial point shown at the top of the diagram. Information is loaded into the unit to the point near the bottom of the diagram. Part of the information is then unloaded. More information may be loaded and unloaded in any manner desired; so that a certain amount of information can be stored between the load and unload addresses. There is instantaneous and simultaneous access to both the load and unload points at any moment. The speed of operation of this unit is high compared with that of the tape unit with which it is associated.

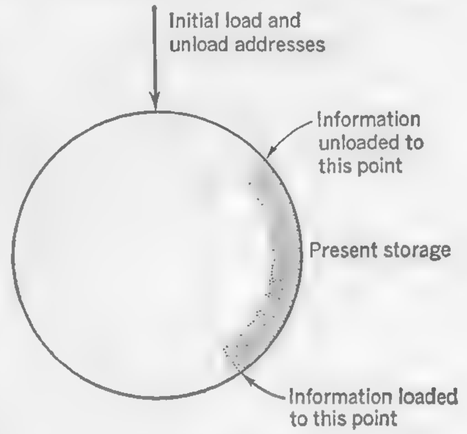

Initial load and unload addresses
Information unloaded to this point
Present storage
Information loaded to this point

FIG. 12.85. Operating-mode diagram of a buffer storage unit.

One derivative of the basic storage unit is a buffer which allows a block of information to be loaded and, at a subsequent time, to be unloaded as a complete block. Units of this type are logically similar to units of the first type; but because they contain less equipment, they can be manufactured more economically. The third type of buffer storage unit is one which performs a major format conversion in the course of its operation. A typical unit in this category is one which reads information directly from the brushes of a punch-card reader and transmits the information in a sequential stream of characters so that it may be written on a paper or magnetic tape. Another unit of this same class performs the converse operation of receiving information as a serial stream of characters and transmitting it in parallel form to a card punch or a high-speed printer. Buffers of this type have achieved a high degree of sophistication. They not only perform the "corner-turning" operation (serial-to-parallel transfer) but also perform a number of logical functions to control the operation of the magnetic-tape unit and of the high-speed printer. They are very nearly complete systems in themselves.

There are a number of other buffer storage units. Some are capable of loading and unloading information in a forward-and-backward direction in order to reverse the information flow. Some perform minor serial-parallel and parallel-serial operations as well as storage operations. In others, sequential and random-access characteristics are combined for the performance of sophisticated format rearrangement.

A Typical Random-access Storage Unit. Figure 12.86 is a block schematic of a low-speed general-purpose random-access magnetic-core storage unit. It is operated by coincident-current techniques and has a storage capacity of four thousand 28-bit words. The unit uses a split cycle, the two halves of which are initiated by sync *A* and sync *B*. If the line labeled "write" is energized at the same time as sync *B*, new information is written into the selected memory cell during the write part of the cycle; if this line is not energized, the information read out of the memory in the first part of the operation is regenerated into the selected memory cell. The timing cycle of the unit is shown in Fig. 12.87. Times shown on this diagram are not the maximum rates of operation of this memory but are the rates at which it is used in conjunction with a computer of a special type. It should be noted that the *X*-set waveform commences earlier than the *Y*-set waveform. Because the memory is used for

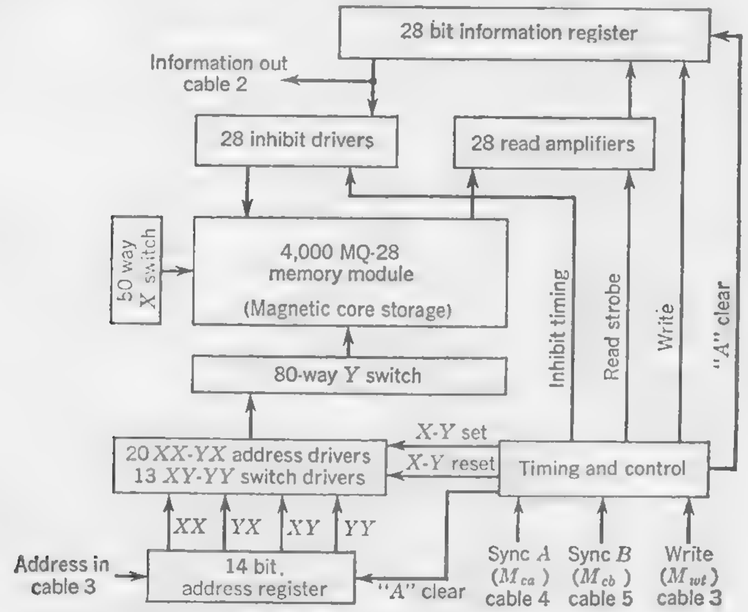


FIG. 12.86. Block schematic of a random-access storage unit.

rather slow applications, extra time may be added to the cycle to improve performance. By starting the *X* current before the *Y*, it is possible to separate the *X* and *Y* noise during sensing and to obtain a better signal-to-noise ratio at strobe time.

The drive system employed in this memory is of the type shown in Fig. 12.73. A combination of nonlinear switch cores and diodes is used to obtain access to the magnetic-core storage array. In Fig. 12.86 the drivers designated as *X*-*X* and *Y*-*X* are the current sinks and those designated as *X*-*Y* and *Y*-*Y* are standard drivers. The latter are actually transistor switches which derive a standard current from current stabilizers. Circuits are so arranged that each driver contains two circuits such as *A* and *A'* in Fig. 12.73. The 50-way switch is driven by 10 sinks and 5 pairs of drivers, and the 80-way switch is driven by 10 sinks and 8 pairs of switch drivers.

Figure 12.88 shows the method of connecting the switches, and Fig. 12.89 illustrates one stage in the assembly of one of the 28 matrices used in this storage unit. The matrices are folded in order to package the 80-by-50 array conveniently.

A Typical Simple Small Buffer Storage Unit. A simplified block diagram of a small buffer storage unit is shown in Fig. 12.90. This unit has a capacity of 144 eight-bit characters. The mode of operation used in the system was described in the section entitled Special-purpose Switch Systems. Loading and unloading can be interlaced as desired, either operation requiring a 10-μsec cycle. The load control and loading switch are used both to drive the storage arrays and, operating as address counters,

to remember the load address. In a similar manner, the unloading switch drives the storage arrays and remembers the unload address. The digit drivers insert information into the storage arrays, and the read amplifiers obtain information from the storage arrays.

There are two ways in which the matrices can be scanned. The first of these is indicated by the matrix and switch arrangement illustrated in Fig. 12.91. The matrix uses a configuration of 9 by 16 and is driven by a 9-way switch and a 16-way switch. Since 9 and 16 are mutually prime, the two switches may count and scan the matrices in a diagonal manner. All positions will be scanned only once. An incomplete scanning diagram is shown in Fig. 12.91*b*. By working out the diagram to completion, it will be found that ultimately all positions are scanned correctly. The buffer actually contains two sets of switches. Provided that these switches

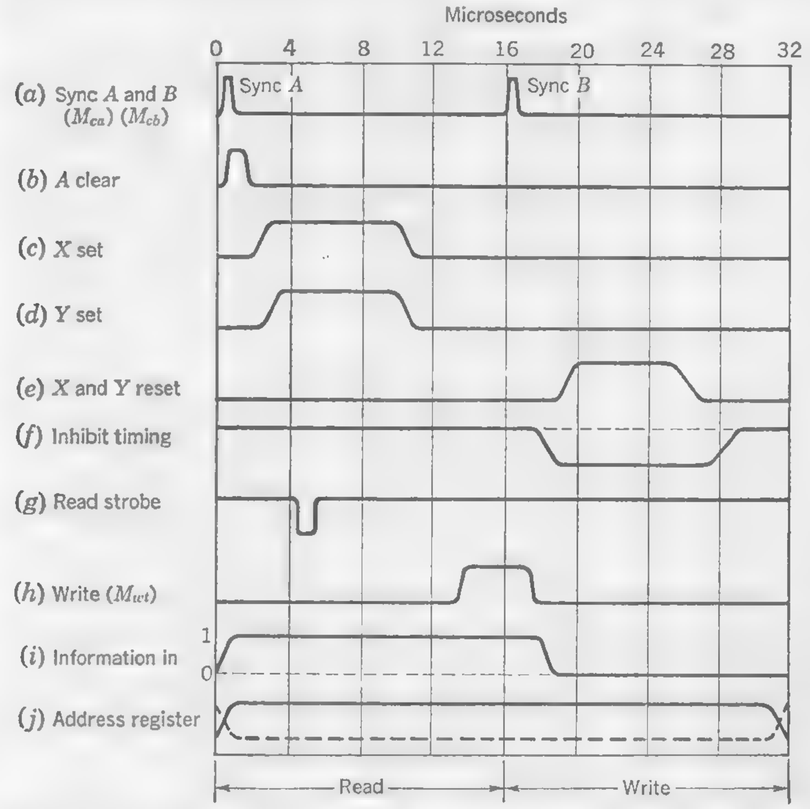


FIG. 12.87. Timing cycle of random-access storage unit shown in Fig. 12.86.

follow the same pattern and start from the same original address, the manner in which the matrices are scanned is unimportant.

Another way of scanning the matrices is indicated by the arrangement in Fig. 12.92. The matrix in this case has a configuration of 8 by 18. Eight and 18 are not mutually prime; therefore, if the mode of scanning indicated in Fig. 12.91 were employed, only half of the 144 cores could ever be scanned. Which half were scanned would depend on the phase of the 8-way and 18-way switches in driving the 8-by-18 matrix. It can be arranged that the 8-way switch generates both phases by connecting each successive pair of cores of a 16-way switch to form 8 outputs. This is the system illustrated in Fig. 12.92*a*. The incomplete scanning diagram in Fig. 12.92*b* shows a "staircase"-shaped scan. All possible positions in the matrix will be scanned only once. As in the previous method, as long as the load and unload switches are of the same type and start from the same origin, this system will operate without indicating the type of scan being performed.

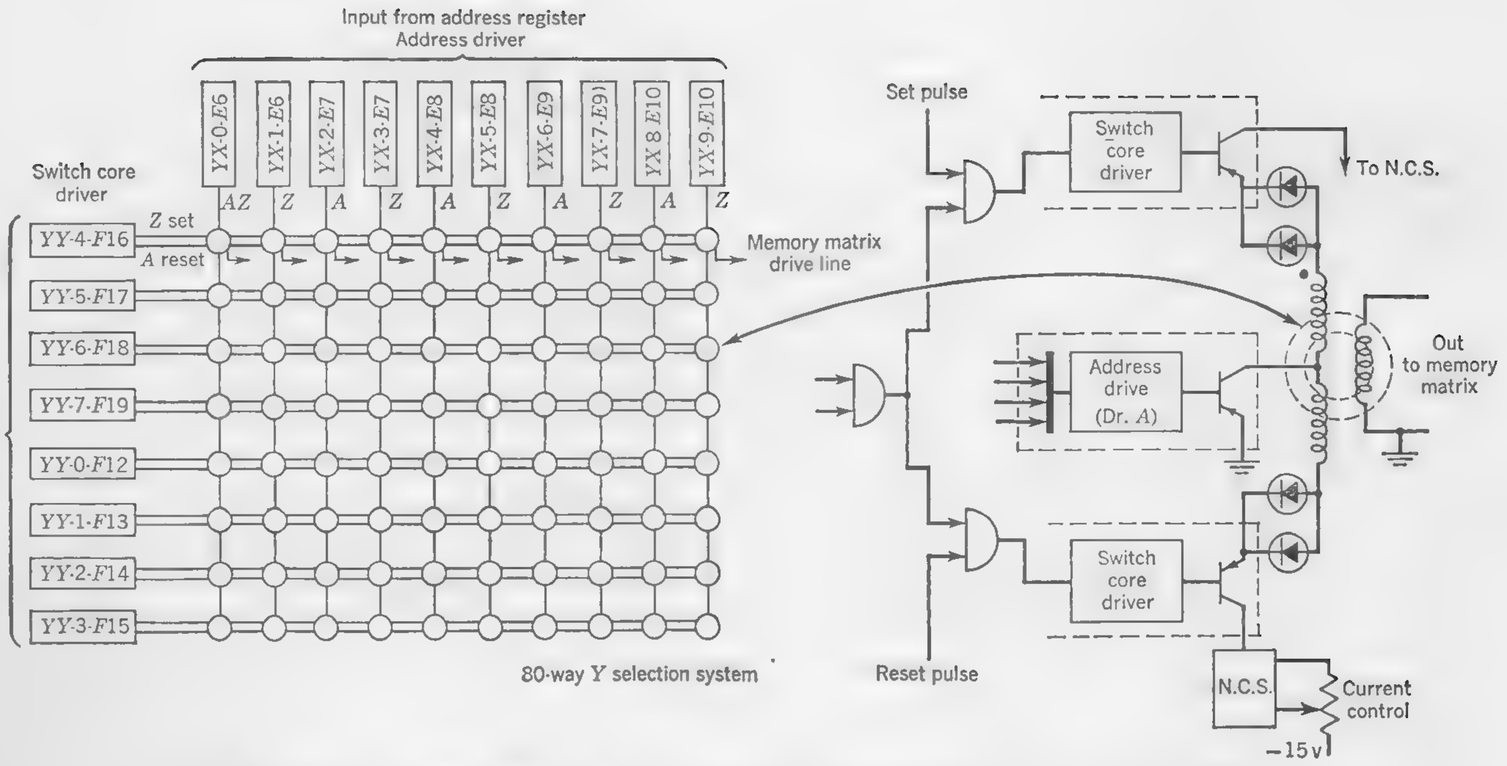


FIG. 12.88. Method of connecting switch drivers in random-access storage unit shown in Fig. 12.86.

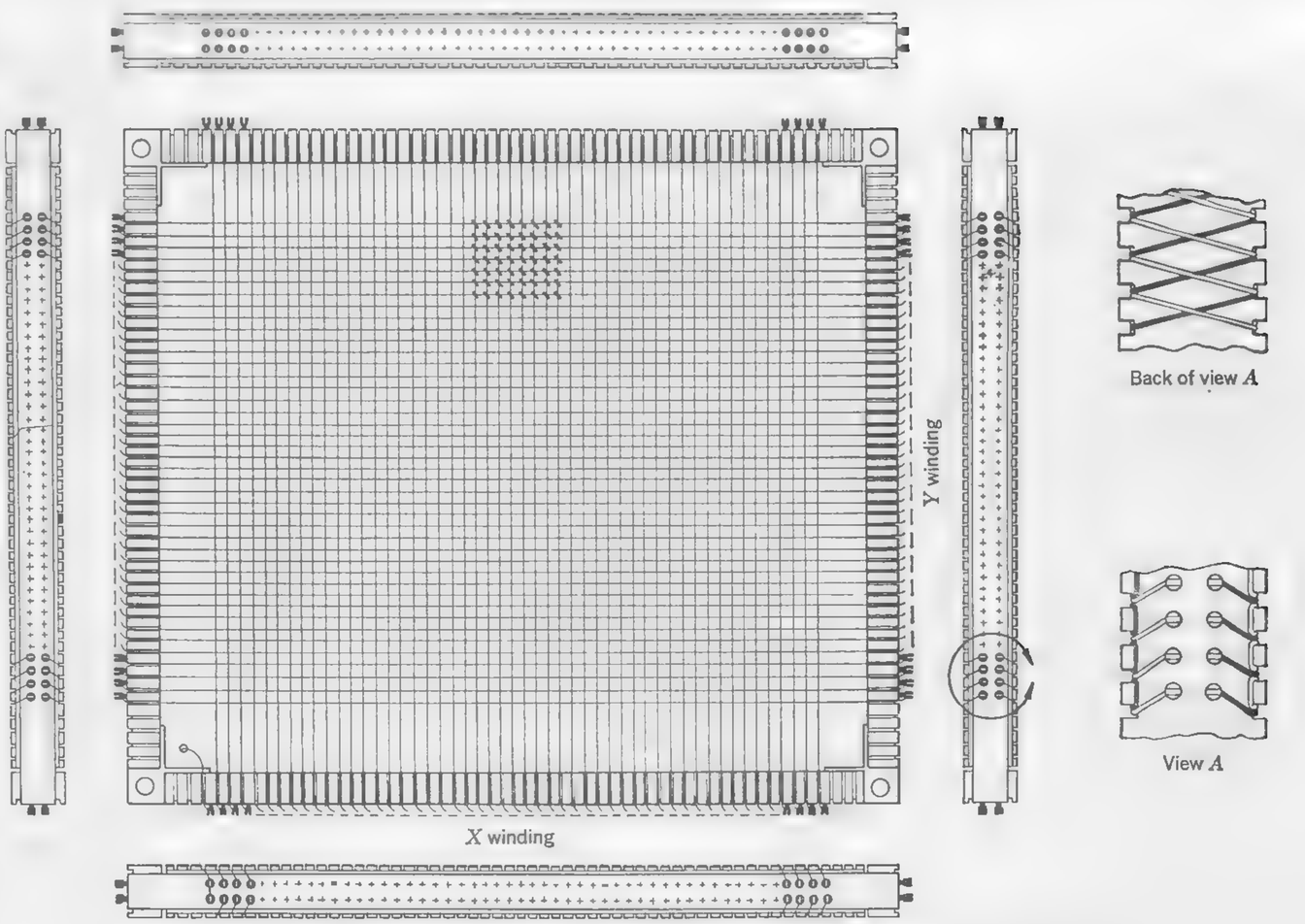


FIG. 12.89. Stage of assembly of a matrix in random-access storage unit shown in Fig. 12.86.

The system shown in Fig. 12.91 is clean and simple to understand. However, it requires a three-phase driver to drive a 9-way switch and a two-phase driver to drive a 10-way switch. In the load control there must be five drivers, three flip-flops, and a number of gates to drive the two switches. The total for the system is 10 drivers, 3 flip-flops, and a good deal of logic. Since the capacity of the storage unit is only 1,152 bits, the number of transistors required to mechanize this many drivers, flip-flops, and logic is unreasonably high. In the system shown in Fig. 12.92 both switches may be driven by a two-phase driver, and these drivers may be in phase. It is therefore possible to use a single pair of drivers to drive through both switches in cascade.

The unit requires only four drivers and two flip-flops and considerably less logic than the system illustrated in Fig. 12.91. Even though the system illustrated in Fig. 12.92 is more expensive in terms of the number of switch cores, there is a considerable reduction in the amount of electronic equipment. It is therefore the preferred driving scheme for a small buffer storage unit.

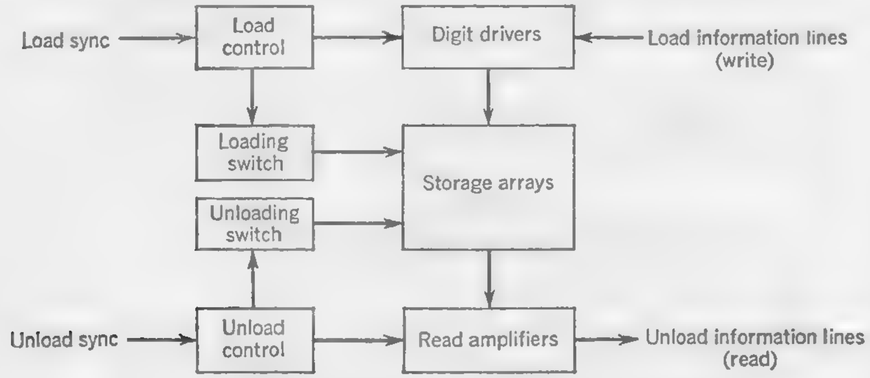


Fig. 12.90. Block schematic of a small buffer storage unit.

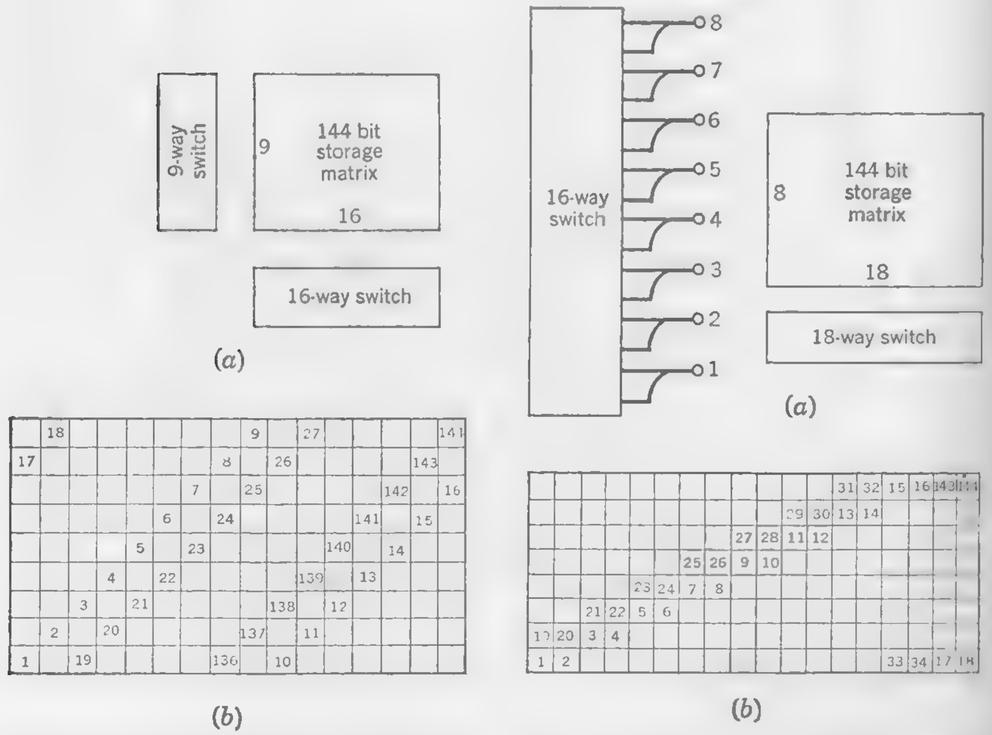


Fig. 12.91. (*a*) Arrangement of switches and matrices in buffer storage unit shown in Fig. 12.90. (*b*) Incomplete scanning diagram.

Fig. 12.92. (*a*) Second arrangement of switches and matrices in buffer shown in Fig. 12.90. (*b*) Incomplete scanning diagram.

Figure 12.93 shows one set of switches and drivers in a 144-bit buffer. There are two identical sets, one of which is used to control loading and the other to control unloading. Two transistor switches are selected alternately by the flip-flop shown at the bottom of the diagram. They steer current originating in a current stabilizer first to the 18-way switch and then to the 8-way switch. The latter is actually a

16-way switch connected in the manner shown in Fig. 12.92. The current passes through the 8-way switch and then becomes the common bus for the 18-way switch. (See Figs. 12.67 to 12.70 for illustrations of the use of this bus.) The 18-way switch then routes the current from the bus into the selected matrix line. The current is gathered together again to form the bus for the 8-way switch, which routes the current through the matrices. Finally the current goes to ground after the second passage through the matrices. In Fig. 12.93 the shaded line illustrates a typical current path at one moment in the over-all cycle of the equipment.

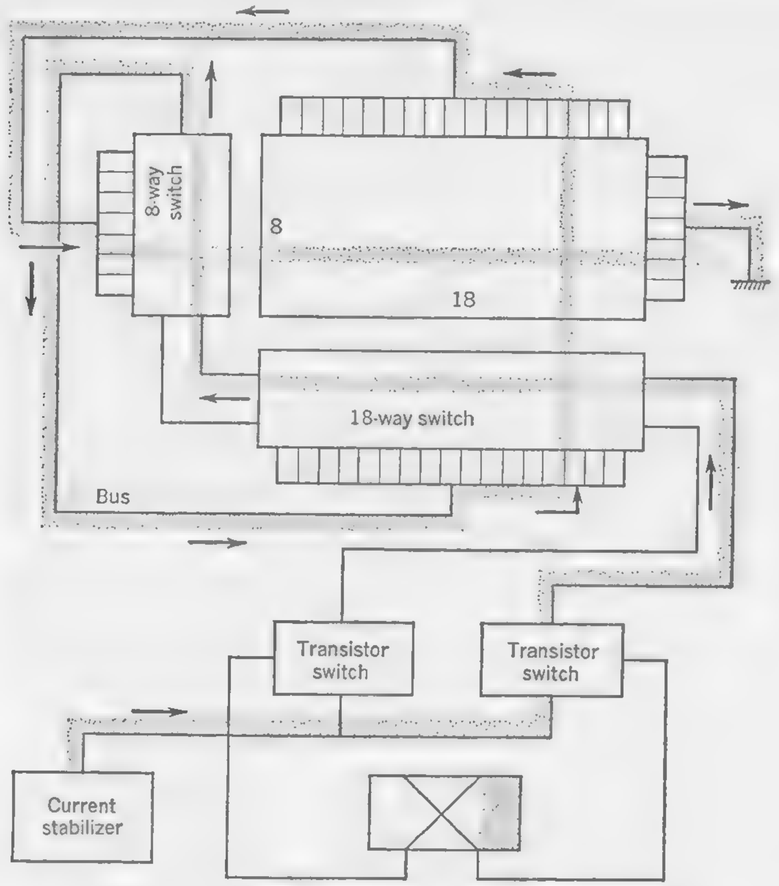


Fig. 12.93. Typical current path in a 144-bit buffer.

Other Features Utilized in Buffer Storage Units. It is possible to use only one set of switches in operating a buffer instead of the two described above. With this mode of operation, a block of information is loaded into the buffer and is then completely unloaded without interlacing of the load and unload operations. The switches are used first to load the cores and are then cleared back to the 0 address in preparation for the unloading operation. Using one set of switches in this operating mode is economical since approximately one-third of the electronics in the buffer storage unit may be removed.

Re-formatting information is a special feature of some buffer storage units. An analysis of the magnetic-core array indicates how this function may be added to the unit. The three dimensions of the core array are conventionally specified so that the X and Y coordinates represent the address and the Z coordinate represents the word length. However, if the three dimensions are considered as axes of reference, it is possible to connect switches to the X and Y coordinates during loading and to the Y and Z coordinates during unloading. Under these conditions, the information

will change its format in the process of passing through the buffer storage unit. This re-formatting is a typical corner-turning operation which occurs in card-to-tape conversions and in tape-to-high-speed-printer conversions. It is also possible to scan the X, Y, and Z coordinates simultaneously by using suitable switches. With such a scanning technique, an input in completely serial form may be read on output in parallel form. This mode of operation is useful in buffers operated in conjunction with teletype equipment where information occurs in both serial and parallel forms in different parts of the system.

The individual switch element in a buffer has both logical and storage properties. Steering-switch elements may therefore be arranged to perform logical operations that are more complex than the counting operation performed in the simple buffer. Additional switches in the buffer unit may be driven from the pair of drivers which perform logical functions in order to control an input or output device.

Partial core planes may also be used to perform special functions within the buffer. One or two cores may be used as markers to indicate when a particular point in the storage operation has been reached. Incorporating such special-function cores often reduces the number of counters required outside the equipment. Special core planes may indicate when a buffer is full or empty, or even half full or half empty. This function is particularly useful in timing the operation of two devices which operate at different average rates of information flow. A typical instance is the conversion of information from one magnetic tape to another. Usually one of the tape units is faster than the other or, in the worst possible case in which variable block lengths of information appear on the magnetic tape, each of the tape units at some time can be faster than the other. Under these conditions, the system must decide whether to stop at the next interblock gap so that the slower tape can catch up to the faster one. The half-empty, half-full type of output signal indicates to the system the approximate state of storage within the buffer. With this signal, the system can choose to stop the tape unit and can determine which of the two tape units should stop in the event that information of variable format is being handled.

A special core plane may be attached to a buffer feeding a high-speed printer. A bit is stored in this extra plane each time a print operation is performed. The signal sensed from the plane can cause another signal to be generated which informs the printer when printing is complete. The printer then stops printing and advances the paper. The signal also informs the magnetic-tape unit that it should start up and feed a new block of information into the buffer storage unit.

There is a multiplicity of functions that can be added to the buffer storage unit quite simply. They are too diverse and specialized in nature to discuss in detail, but in general most of them can be designed by a serious study of the magnetic-core art.

12.7.19. System Design of Coincident-current Core-storage Units. Core-storage units may vary greatly in size and application. A basic requirement in achieving a highly reliable and economical storage unit is an effective system design. In designing a core-storage unit, one of the greatest difficulties is the initial formalization of the specification. As much as one-fifth of the time required for design and manufacture of a prototype unit may be devoted to formalizing the specification.

The designer should be guided by the following considerations in laying out a system design:

1. A system should incorporate the least number of components necessary to perform its series of operations correctly. Careful investigation of the system design at the component level is essential.

2. In general, the cost per bit of a core-stage unit decreases as the capacity is increased and as the speed is decreased.

3. The amount of timing and control equipment is nearly constant in all sizes of storage units.

4. The number of address circuits per word stored decreases as the number of words increases.

5. An increase in the amount of equipment associated with digit-plane drivers, read amplifiers, storage registers, etc., is usually required as the word length is increased.

6. The amount of equipment increases very rapidly as the speed of operation is increased.

7. Core materials with long turnover times are more economical in operation than those with short turnover times. The cycle time in a system includes not only the storage operating time but also time for "red tape" (setting up addresses, waiting for information to come down from the memory register, propagation delays). In a medium-speed memory, the actual operating time of a material with a 1-μsec turnover time is 3 to 4 μsec, while the red tape requires 3 to 4 μsec. If a material with twice the turnover time is used, the memory operation requires 6 to 8 μsec and the red-tape time remains 3 to 4 μsec. When a 4-μsec material is used, the operating time is about 12 μsec and the total cycle time is 16 sec.

8. A sequential-access memory is more economical to operate than a random-access memory. Even though core matrices may be of the random-access type, sequential or semisequential access to the addresses is feasible.

Designing a system for a high-speed parallel-storage unit presents no unusual problems. The cores are operated at maximum speed, and the number of parallel read amplifiers and digit-plane amplifiers corresponds to the number of parallel bits in the words stored. In some system designs, several storage units are required. They are arranged for interlaced operation to avoid time delays for regeneration in each unit. For example, two 8-μsec units may be made to operate as one 4-μsec unit. If all odd-numbered addresses are ascribed to one storage unit and all even-numbered addresses to the other, the system tends to use the units alternately.

In data-processing systems, the flow of information is often in the form of 4- to 7-bit parallel characters rather than in words of fixed length. The characters frequently flow at the rate of 4 to 5 μsec per character. Two methods have been adopted for transmitting information in this format. The first method is suitable for storing words of variable length. One word is divided into subgroups of four to seven characters each. A 20- to 35-bit parallel-storage unit with converters at the input and output will store the words in their appropriate character format. The second method is used for storing words of fixed length. Two or more parallel-storage units of 20 to 35 bits each together store the complete word. The cycles of these units are interlaced so that information can be transferred either to or from the complete storage unit in a continuous flow.

Occasionally a very long word must be transmitted in a parallel manner, but at a rather slow rate. In this case, the storage unit will take one-half or one-quarter of the word length at a time. Information is read out in two- or four-character sequences and is then shifted into a long parallel register which presents the word in parallel to the computer.

Serial computing machines often employ a partially parallel storage unit. In designing a system of this type, the designer must try various numerical combinations of address circuits and of digit-plane and read-amplifier circuits in order to achieve the proper balance as well as maximum economy. The fact that storage units are usually parallel, even in serial machines, can be used effectively in most computer designs. Since instructions and addresses are usually required in parallel, they are transferred in a parallel manner to the computer while numerical information is transferred in a serial or semiserial manner. Such an arrangement can increase the computing speed considerably.

In applications involving electromechanical devices, such as tape transports, printers, and card feeds, the storage unit may be required to operate in a slow sequential manner. Because sequential-storage operation is more economical than random-access operation, it may be advisable to make minor changes in a system design to utilize sequential units. In units of large capacity the economical advantages of sequential operation are less important than in smaller units. The major cost in large units is represented by the magnetic cores, drivers, and sense amplifiers, and not by the controls. Large storage units usually have completely random properties, which can be utilized to improve the design of the computing system.

One feature of system design requiring careful attention is ease of maintenance of the equipment. The design must include provision for easy access to the components,

plug-ins, suitable monitoring points, and other items of this type. A marginal checking arrangement should be provided. It should not be so complicated that the maintenance engineer must spend a great deal of time performing marginal checks on the system. If ample time is expended on solving maintenance problems at the initial design stage, it is likely that the cost of long-term maintenance will never exceed the procurement cost of the equipment.

12.7.20. Design of Core-storage Units for Adverse Environments. Three major problems are encountered in designing a magnetic-core storage unit to operate under adverse environments. First, it is necessary to design circuits which can operate reliably under these conditions. At present the circuit elements represent the major limitation on the extremes of environment for which the design can be effectively completed. Secondly, it is necessary to design the circuits and magnetic-core portions of the equipment to operate satisfactorily over a wide range of temperature. And, thirdly, it is necessary to design a unit to withstand considerable shock and vibration.

The problem of designing adequate circuits is not peculiar to the magnetic-core storage unit, but special conditions are imposed by this unit in that the drive circuits must be able to handle large currents and large voltages under extremes of temperature and vibration. The best solution is to design the drivers so that they cannot operate continuously but must operate in an intermittent pulse mode. By this means, dissipation in the unit can be reduced sufficiently for adequate operation.

Operation of a core-storage unit through a wide range of temperatures presents a problem which the designer may solve by one of three approaches. The first is to enclose the cores in a temperature-controlled box which will operate at the maximum temperature encountered in the system. A disadvantage of this technique is that time is lost in heating the box to the correct temperature before operation of the storage unit can begin. A second approach is temperature compensation of the entire system to adjust the current automatically to the correct value for any temperature encountered during operation. A third technique which has been successful is to combine heating of the core-storage unit and temperature compensation. This approach allows the equipment to be started at a low temperature and to be compensated continuously while the storage unit is being heated. When the final operating temperature is achieved in the core matrices, the equipment then operates as a temperature-regulated system. The advantage of this compromise approach is that the system can be switched on quickly and will operate instantaneously.

It has been found that, if information is stored in a core which is cycled to extremes of temperature of about 100°C, the flux state in the core is decreased by approximately 1 or 2 per cent for each complete temperature cycle. This loss of flux is significant only if the storage unit must retain its information without regeneration through many temperature cycles. Since this condition is encountered rarely, a temperature-compensation scheme is usually the most adequate approach to the temperature problem.

Under conditions of excessive vibration or shock, the conventional open-wired type of matrix construction is ineffective. Cores should not be encapsulated in a solid plastic because the stress applied to them at the point of contact with the plastic changes their electrical characteristics materially. Furthermore, cores should not be constricted during their electrical operation because magnetostriction changes the size of a core abruptly during the transition from one flux state to another. Cores must be cemented or encapsulated in a flexible compound which allows them freedom of movement during magnetostriction and which does not impose any stress on them during the encapsulation process. Satisfactory matrices may be constructed by two techniques.

The first is to cement each core with a flexible plastic into a small hole in a plate. The plate should be made of aluminum or some other highly conductive metal. It is particularly important to keep the stack of cores at a uniform temperature to ensure that noise is not introduced by temperature variations in the stack. The cores are then wound by a sewing technique rather than the usual 45-deg winding. This sewing technique is more expensive but has the advantage that the entire aperture of the core is available for windings rather than the small ellipse available with the 45-deg type of winding. Much smaller cores may be used in this array construction than in an ordinary mat construction. Using cores a fraction of a milligram in mass, it is possible to fabricate a very small, compact storage array which can stand large amounts of vibration and shock.

The second technique is to cement conventionally constructed mats with a flexible cement onto aluminum or other conductive plates. The plates and their assembled mats are bolted together in a package, to which terminals and interconnecting wiring are added. The whole unit is then impregnated with a suitable plastic to form a solid, rugged core stack. Arrays constructed by this second technique are cheaper but less reliable under conditions of excessive vibration and shock than the smaller, lighter arrays constructed by the first technique.

Mechanical stress on cores alters their electrical properties. In fact, high vibration or shock can render a core-storage device useless. Particular care must be taken in the design to ensure that minimum resonance is associated with the magnetic portion of the equipment. The flexible cement used in encapsulating magnetics must be of a type which reduces the impact under shock.

Although magnetic cores are not completely ideal elements for use in adverse environments, they introduce fewer problems than most other storage mediums or even most electronic devices. The techniques described above enable a designer to devise units which will operate reliably under extremes of environment.

12.7.21. Testing of Magnetic-core Storage Units. There are two types of tests that must be performed on a magnetic-core storage unit. The first are design tests which are performed to ensure that design specifications have been met completely and properly. The second are tests that are performed on the production units.

Most of the design tests are conducted throughout the period of design of the equipment with the intent of achieving maximum tolerance for each component. The final design tests are made under conditions which simulate as closely as possible the final operation of the equipment. It is rarely sufficient when conducting these tests to connect the memory unit to the computer, because of the difficulty of programming the computer to simulate worst-case operating conditions. In actual operation a computer can generate a condition that is much worse than a calculated worst-case condition. For this reason, the design-acceptance tests should be highly flexible and very stringent. They should aim for performance under conditions that are more severe than the predicted worst-case conditions. Thus the designer can be reasonably sure that unpredictable worst-case conditions encountered in using the memory will not cause failure of the equipment. Design-acceptance tests should include temperature tests, marginal tests, and as many program checks as can be performed. It is advisable to use a special testing device rather than the computer for performing design tests. With the special tester, it is possible to program rates of operation in excess of computing rates of operation and also to program very slow rates, which are difficult to attain in the average computer.

The production tests are of an entirely different nature from the design tests. They start with incoming inspection of the components. All components are rigorously tested to ensure that they have no areas of incipient failure and that they are within the design tolerances. Every magnetic core used in the equipment is checked against a well-specified set of test conditions. The components are then combined into basic subassemblies, which are tested thoroughly for possible damage during the assembly process. Present design practices allow for large end-of-life tolerances in components. For this reason, an operational test of the basic subassembly is usually insufficient. An adequate test can be performed only on the individual components to ensure that they are still within tolerance after subassembly. The individual core matrices, for example, are rarely tested by dynamic methods. Each core in the matrix is tested individually to ensure that no more than a maximum specified degradation has occurred in the assembly process. At the same time, the windings are checked to make sure that every winding links every core in the correct direction. The purpose of the tests is not to determine whether the subassembly works but to check that the components have not been damaged. In a similar manner the frame of the memory is checked to ensure that all wiring is correct and that any components included as part of the frame have not been damaged during assembly.

Theoretically, at this point all subassemblies could be plugged into the main frame

and the unit would work perfectly. If the unit is tested and found inoperable, the quality-control procedure up to the point of final assembly is in error. If the system design incorporates very large end-of-life tolerances, it should be difficult to cause the new equipment to fail during initial testing. A final series of marginal and functional tests on the completed unit are usually performed under adverse environmental conditions.

12.7.22. Maintenance of Coincident-current Storage Units. Maintenance of early tube-driven core-storage units was excessive. To reduce the time and simplify the techniques for maintaining equipment, an engineering-maintenance panel was added to each unit at considerable cost. High computational reliability could be achieved only with specified maintenance once in 8 hr or, for some units, once in 24 hr. Most of these early tube-driven units that were well designed and maintained properly are still operating with an error probability close to 1 per 1,000 hr of computing time.

The introduction of transistor drive and the improvement in tolerancing and design techniques have contributed to the development of storage units of much higher reliability than the earlier equipment. Maintenance requirements of the newer storage units may be divided into three categories. First-level maintenance is applied to large random-access coincident-current units operated in conjunction with data-processing equipment. The second level is required for small units or buffer storage units used in special-purpose computing equipments. Third-level maintenance is used for very small units, usually buffer storage units, combined with computing or other digital equipments.

First-level Maintenance. The storage unit is equipped with facilities which allow changes in supply levels and operating currents for purposes of marginal checking. Programming the equipment for marginal operation is accomplished by generating an information flow into the storage unit most likely to cause failure. If a unit fails during a scheduled marginal check, the component which caused failure is immediately replaced. The frequency of periodic first-level maintenance and the type of tests performed depend on the reliability of the equipment design and will vary widely for different types of equipment. Large units need more frequent maintenance than small ones. New equipment requires a short period between scheduled checks, which may be gradually lengthened as the equipment ages. Scheduled maintenance about every 50 hr is advisable for most new equipments; for older equipments, a schedule of checking once in 200 to 500 hr is adequate.

Second-level Maintenance. Second-level maintenance is applied to small, rather reliable units which are incorporated in moderately complex special-purpose data-processing devices. Since devices of this type usually are operated in an "off-line" manner, parity checks are performed while information flows through the equipment. The storage units are equipped with rather primitive means for marginal checking. They may also be equipped with a simple test device which permits checking of the unit without operating the whole equipment.

Maintenance is performed infrequently, usually at a periodicity such that the probability of error between scheduled maintenance periods is relatively high. A probability of 0.2 is considered sufficient. Maintenance may be scheduled only once in 3 months to a year. Periodicity of maintenance and the nature of the tests that are performed vary widely for different types of equipment, depending on their size and the general reliability of their design.

Third-level Maintenance. Units which are subjected to third-level maintenance are very small and highly reliable. Under their operating conditions, scheduled maintenance is difficult to perform and an occasional error may be tolerated. Often an error-checking device may be used to indicate when the frequency of errors within the unit has risen to a dangerous level.

Third-level maintenance is not scheduled maintenance at all. Whenever observation of the device indicates the occurrence of a large number of errors, maintenance is performed to eliminate the defective component or subassembly. Frequently at this level of maintenance, no attempt is made to replace a component that has failed; instead the defective subassembly is located and replaced by a new one. Such an approach to maintenance is not very costly because units of this type are so reliable that the probability of failure is less than once in 6 months to a year.

Maintenance of Future Storage Units. The magnetic-core-storage art is rapidly approaching the point at which a fourth level of maintenance can be introduced. If a storage unit is so inherently reliable that the whole package may be considered a single subassembly, in the event of failure the entire device may be discarded and replaced by a new one. This type of maintenance can be economically introduced when the unit is small and when the frequency of failure is once in 3 to 5 years. At present, the frequency of failure in transistor-driven storage units is once in 6 months to 2 years in equipment containing 100 to 300 transistors per unit. Highly reliable core-storage units with fewer than 50 transistors per unit will undoubtedly be designed in the near future. A probability of failure of less than once in 3 to 5 years can be expected of these future units. At that time, fourth-level maintenance will be more economical than having available trained personnel who can perform the three other levels of maintenance.

12.7.23. Nondestructive Read-out. One of the disadvantages of coincident-current operation of a core-storage unit is that information is destroyed during reading and must be rewritten. Reading time is long and loss of information, during transfer from an electronic register and back to the core matrix, may occur. In most computing applications, reading occurs more often than writing. Therefore, the shortened reading time implicit in nondestructive read-out would greatly increase computational speeds. Another disadvantage of coincident-current operation is that information is read out as a single short pulse. Nondestructive read-out would allow repeated interrogation of a core to provide a continuous output.

If a nondestructive reading method is to replace the present destructive method, it should satisfy the following requirements:

1. Reading time should be short.
2. Writing time should not be much longer than that required by presently used methods.
3. The method should be adaptable to storage units of large capacity.
4. The fabrication cost of a nondestructive system should not be much greater than that of destructive systems.

To date no method which is likely to meet all four requirements has been proposed. Three methods which will be useful in restricted applications are r-f sensing, read-out based on the quadrature-field principle, and read-out using cores with secondary apertures. It is too early to evaluate the merits of these three approaches to nondestructive read-out.

R-F Sensing. Radio-frequency sensing is a nondestructive method of sensing the static state of the magnetic field in a core. The characteristics of the minor loops of a core are different for the 0 and 1 storage states. A phase change or inversion, a beat frequency, or other similar effect is used to detect the state of a selected core. The disadvantages of using radio frequencies for read-out are that the technique operates on second-order characteristics and that it has not yet been found successful in interrogating core matrices of any reasonable size.

A pulse method for accomplishing nondestructive read-out has been developed recently. Two cores connected in opposition to the sense winding are used for each bit. This arrangement cancels the first-order characteristics and allows the secondary effects to be sensed directly. In order to interrogate a core, a low-current pulse of short duration (a few millimicroseconds) is applied to the core pair. A short pulse is obtained in the sense winding. Depending on the method of connecting the cores, it is possible to differentiate between states either by the polarity of the sense-winding output or by the presence or absence of a pulse in the sense winding. At present this technique can be used only in linear-selection storage units of small capacity.

Read-out Based on Quadrature Field. Figure 12.94*a* shows a core which is saturated in a direction around the toroid. A solenoid tends to magnetize the core in a direction perpendicular to the original direction of magnetization. Because the material is already in a saturated state, it cannot be further saturated. As shown in the vector diagram in Fig. 12.94*b*, the resultant vector is not the vector sum of the two fields but

is the original vector rotated through an angle to form the quadrature field. The rotation causes a reduction of flux around the toroid. If the field was in the opposite direction originally, there would be an algebraic increase of flux. The output of a sense winding through the core is a positive pulse for a 1 and a negative pulse for a 0. When the quadrature field is released, the core returns to its original state.

A system operating on the quadrature-field principle is truly nondestructive. Cycle times are short and the output pulses produced are of large amplitude and opposite polarities. It is possible that a nondestructive read-out technique based on quadrature field may be developed from the dipole-switching mode used to operate some experimental thin-film storage devices. At present, the switching rate of cores used in this mode is so high that it has been impossible to produce pulses of sufficiently short duration to disturb the cores without switching them.

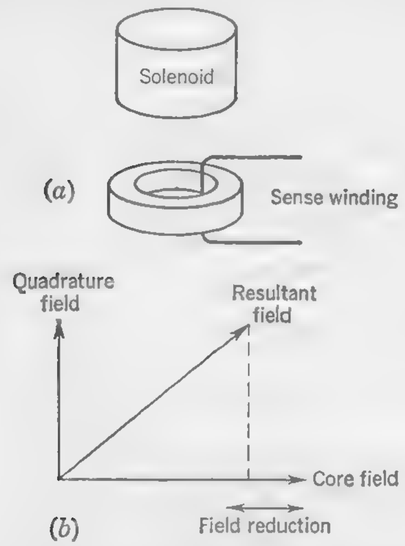


FIG. 12.94. (*a*) Toroid being magnetized by a solenoid. (*b*) Vector diagram of quadrature field.

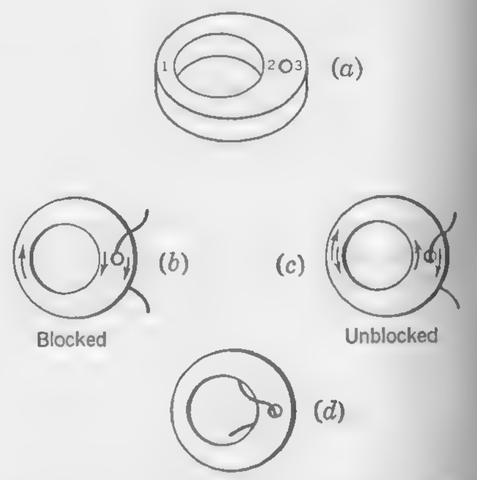


FIG. 12.95. (*a*) Flux paths of core with small and large holes. (*b*) Blocked core. (*c*) Unblocked core. (*d*) A method of unblocking core.

Recently special ferrite cores have been manufactured with two apertures in orthogonal directions. One aperture is used in a conventional storage manner, and the other is used to produce the quadrature field. Although these cores now require large operating currents and are relatively complex to manufacture, they offer considerable promise for a satisfactory nondestructive read-out technique in the future.

Read-out Using Cores with Secondary Apertures. A nondestructive technique may be used to interrogate a storage core which has several possible flux paths. A core with an additional small aperture will have more flux paths than one with a single aperture. For example, there are three flux paths in the core illustrated in Fig. 12.95*a*. The area of path 1 must be greater than the sum of the areas of paths 2 and 3. A flux path encircling the small aperture must be much shorter than a flux path encircling the large aperture.

A large mmf is applied to the core by means of one or more windings through the large aperture only. The entire core is saturated in one direction as illustrated in Fig. 12.95*b*. A winding through the small aperture can cause no change of flux around that hole unless the applied mmf is large enough to change the flux in path 1. A sense winding through the small aperture will produce no output when the core is in this state, and the core is said to be blocked. If an mmf in the reverse direction to the original blocking mmf is then applied by means of windings through the large hole, the field in the core will tend to be reversed. If the second mmf is increased slowly, there will be no action until a threshold value is reached. Then part of path 1

and all of path 2 will be reversed, and after another threshold, path 3 will be reversed. The first threshold is caused by rectangular characteristics of the core. The second threshold is caused by the increase in mmf needed to switch path length 1, 3. Because path length 1, 2 is shorter than path length 1, 3, more mmf is required to switch path 1, 3 than is required for path 1, 2. These parameters can be adjusted by varying the material and geometry of the core. The thresholds in paths 2 and 3 can be far enough apart for all practical purposes. When path 2 has been reversed and path 3 has not, the core is said to be unblocked (Fig. 12.95*c*). If an alternating mmf is applied by means of a winding through the small aperture, an output will appear on a sense winding through that aperture. The direction of magnetization in paths 2 and 3 will reverse continuously, but the unblocked condition of path 1 will not be changed by the interrogating action.

The core can also be unblocked by an mmf applied to a winding linking both holes (Fig. 12.95*a*). A large mmf is needed in order to reverse the path length between 1 and 2. There is only one threshold with this type of winding. The direction of magnetization of paths 2 and 3 cannot be reversed.

Since both the large and small apertures have storage-type thresholds, it is possible to wind cores of this type in a double coincident-current matrix. One matrix links the large holes of all the cores, and the other matrix links all the small holes. The first matrix is used for writing and the second for reading. In order to write, currents are applied to the selection wires of the write matrix and the cores are blocked. Then, if desired, current is applied in the reverse direction and the cores are unblocked. The blocking current should be somewhat larger than the unblocking current. In order to read, coincident currents are applied to the read matrix in a direction which will reverse the magnetization around the small hole, and then the currents are reversed to restore the original direction of magnetization.

A system of this type is truly nondestructive, but it has three disadvantages. First, the cores required must be larger than those used in destructive systems and, consequently, the power required for a writing action is increased. Secondly, it is necessary to wind the cores in a matrix consisting of a double mat of wires and to drive the mats separately. This leads to an increase in costs. Thirdly, read-out time is longer than that of a quadrature-field system because two pulses are required for each output. Furthermore, the device operates as a result of flux transference between paths 2 and 3 with consequent slow domain-wall movement. The minimum read-out time is about twice the characteristic switching time of the material plus red-tape time for setting up the reading matrix. In practice, complete flux transference between paths 2 and 3 is not necessary. Operating time can be shortened by employing noncoincident current in the reading and writing cycle. Using very large noncoincident currents for interrogation, it is possible to sample the core in about one normal switching time of the core material. A 1-μsec material, for example, can be interrogated at the rate of 1 Mc.

Nondestructive read-out with secondary-aperture cores has been successful in small storage devices where continuous access to stored information is essential. More significant is the application of this technique in the work being done on transfluxors and multiaperture devices (MAD). This work has contributed to the development of highly sophisticated, all-magnetic forms of logic and shift registers.

12.7.24. Future Core-storage Units. The development of the magnetic-core storage unit has been in the direction of faster operation and larger capacity. It can safely be predicted that storage devices will operate at a cycle time of much less than 1 μsec within the next few years. These devices will find applications in millimicrosecond computers, which will probably be the next generation of scientific machines. Data-processing machines will require core-storage units of large capacity which can be driven by low power. Power requirements have been markedly decreased since the first storage units were introduced. Present transistor-driven units require one-tenth the power required in the earlier tube-driven units. Research is in progress to design magnetic-core drivers which are controlled by lower-powered transistors than those in use today. Future units will probably need no power transistors but will rely on high-frequency a-c lines for driving the magnetic elements. Only control

functions will be performed by transistors. Core-diode-transistor logic which has greatly reduced the number of transistors required in a storage unit has already been developed.

A continuous trend in the design of core-storage units is the reduction of physical size. Initially units were excessively large and unsuitable for use in mobile equipment. By developing more closely packed matrices, designers have been able to produce very compact units. Storage units employing a packing factor in excess of 50,000 bits per cubic foot including address circuits and memory register are now available. Considerably higher packing factors can be achieved in the future.

Future core-storage units will probably be either fairly large, very high-speed devices requiring many transistors or large-capacity, compact, slow, reliable units requiring few transistors. There will undoubtedly be intermediate units of medium capacity which will be highly reliable and compact for mobile use.

12.8. FERROELECTRICS AND THEIR MEMORY PROPERTIES*

Because ferroelectrics are dielectrics, a condenser containing ferroelectric material as a dielectric can be electrically polarized by an applied electric field. If polarization is plotted against applied field, a hysteresis loop is observed similar to the hysteresis loop of ferromagnetic materials. Ferroelectric hysteresis is due to the spontaneous alignment of dipoles which can be reversed by reversing the applied field. Polarization observed from the hysteresis loop when the applied field E is zero is remanent polarization P_r. Remanent polarization binds and retains charges in a small storage condenser for a long period of time without appreciable loss even if the electrodes of the condenser are short-circuited during this period. This fact is very important because it permits the use of simple switching circuits, as well as the omission of regenerative operations.

Memory properties of ferroelectrics may then be described in terms of remanent polarization, characterized by the detailed features of the dipoles which can be reversed in an applied field by a small displacement of certain atoms held by combined binding forces of ionic and homopolar character. Presence of an electrical activation energy, best observed by the dielectric hysteresis loop, is the distinct characteristic of ferroelectrics.

Ferroelectric properties have been discovered in a wide variety of dielectrics. The first ferroelectric material discovered, rochelle salt or seignette salt, $KNaC_4H_4O_6{\cdot}4H_2O$, is commonly known as tartrate.

A second group is represented by alkali-metal dihydrogen phosphates and arsenates such as KH_2PO_4 and $KD_2A_3O_4$. Barium titanate, $BaTiO_3$, is an important representative of a third ABO_3 type of compound and, at present, is most promising for practical application. This group is also known as the oxygen-octahedra compound group, which includes potassium niobate, another very promising member of this family.

Compounds such as $Cd_2Nb_2O_7$ and $Pb(NbO_3)_2$ have been found to be ferroelectric; these, however, are only two examples of more general groups such as the $A_2B_2O_7$ type. Among these new compounds, guanidine aluminum sulfate hexahydrate (and its isomorphs), $C(NH_2)_3Al(SO_4)_2{\cdot}6H_2O$, with rather complicated molecular structure was found to be the parent of a new family of ferroelectric materials.

Table 12.30 presents the basic oxygen-octahedra type of ferroelectric materials while Table 12.31 contains recently discovered ferroelectric materials. The tartrates, as well as the alkali-metal dihydrogen phosphates and arsenates, are not listed because they have not been significant in this field.

Spontaneous polarization P_s is the electric moment per unit volume when all the electric dipoles are aligned in the same direction.

Coercive field E_c is the field strength E at which approximately half the electric dipoles have been reversed from a condition of complete saturation. The net polarization is zero at this point. E_c is dependent upon temperature as well as frequency, amplitude, and waveshape of the applied alternating field.

* By Charles F. Pulvári.

Curie temperature T_c is the temperature beyond which a ferroelectric material does not exhibit a hysteresis loop. For more detailed definitions, reference is made to the definitions of electric characteristics of ferroelectric materials and devices published by The Institute of Radio Engineers, Inc.

12.8.1. $BaTiO_3$ Single Crystals. At present, barium titanate single crystals show the most favorable properties for storage and switching applications. The crystals are artificially grown from a melt of barium titanate in a potassium fluoride flux.

Table 12.30. Ferroelectric Materials of the Oxygen-octahedra Type

	T_c, °C	Spon. polarization 10^{-6} coulomb per cm^2	Coercive field, volts per cm
Titanates:			
$BaTiO_3$	120	26	1,200
$PbTiO_3$	490	.....	.....
$SrTiO_3$	(−56)–(−269) (approx)	1–3	.....
$CdTiO_3$	−223		
$CaTiO_3$			
$FeTiO_3$, $NiTiO_3$	700 (approx)		
Niobates:			
$KNbO_3$	435	0.9–3.8	.....
$NaNbO_3$	−110	.....	.....
$LiNbO_3$	Similar to $LiTaO_3$ data shift toward higher T	.....	.....
$Cd_2Nb_2O_7$	−88	1.8 at −173°C with 25 kv per cm	.....
$Pb_2Nb_2O_7$	−258	.....	.....
$Pb(NbO_3)_2$	570	Not saturated under 60 kv per cm	.....
Tantalates:			
$NaTaO_3$	475	.....	.....
$KTaO_3$	−260	.....	.....
$LiTaO_3$	.....	1–23	6,000
$Pb(TaO_3)_2$	260		
$RbTaO_3$	247		
Stannates:			
$PbSnO_3$	150	.....	.....
$(Ba,Sr,Ca)SnO_3$	120–(−165) (approx)		
Zirconates:			
$PbZrO_3$	230	.....	.....
Vanadates:			
$NaVO_3$	.....	.....	.....
Rhenium trioxide:			
WO_3	710	.....	.....
MO_3	(Higher than) 530	.....	.....
Gallates:			
$LaGaO_3$	475 (approx)	.....	.....
Ferrates:			
$LaFeO_3$	?	.....	.....

A method for growing $BaTiO_3$ single crystals was described by J. P. Remeika.[69] It crystallizes as pairs of flat plates, resembling butterfly wings. The angle of the twins with a common 110 edge is approximately 40 deg. Under closely controlled conditions, it is possible to produce perfect crystals.

12.8.2. The Bistable Storage Cell. For practical uses ferroelectric materials with nearly rectangular hysteresis loops and low-coercive-force fields are preferred. When perfect c-domain single crystals are used as a dielectric in a storage cell the hysteresis

Table 12.31. Other Ferroelectric Materials

Substance	Formula	T_c, °C	Spon. polarization 10^{-6} coulomb per cm²	Coercive field, volts per cm
Guanidine aluminum sulfate hexahydrate	$C(NH_2)_3Al(SO_4)_2{\cdot}6H_2O$	About 300	0.35 at 23°C	1,500
Methyl ammonium aluminum alum	$(CH_3NH_3)Al(SO_4)_2{\cdot}12H_2O$	−97	0.6 at −107°C	.....
Urea chromium alum	$(CONH_2NH_3)Cr(SO_4)_2{\cdot}12H_2O$	−113	0.1	.....
Ammonium sulfate	$(NH_4)_2SO_4$	−49	0.25 at −58°C	.....
Ammonium cadmium sulfate	$(NH_4)_2Cd_2(SO_4)_3$	−170	0.3 at −90°C	.....
Colemanite	$Ca_2B_3(OH)_3{\cdot}H_2O$	−2	0.6	.....
Thiourea	$(NH_2)_2SC$	−104	3	1,000
Glycine sulfate	$(NH_2—CH_2—COOH)_2 + H_2SO_4$	47	2.4 at R.T.	200
Glycine selenate	$(NH_3CH_2COOH)_2SeO_4$	22	.....	.....
Heavy glycine sulfate	$(NH_3CH_2COOH)_2SO_4$	60	.....	.....

loop is practically a rectangle, as shown in Fig. 12.96. Maximum value of P_r is obtained from a saturated hysteresis loop.

In plotting polarization P instead of displacement D, dielectric displacement in air caused by applied field E is neglected since its value is only 0.1 per cent, approximately, of remanent polarization P_r in the case of a nearly rectangular loop. After P_r, coercive-force field E_c is the most important characteristic of a P-E loop.

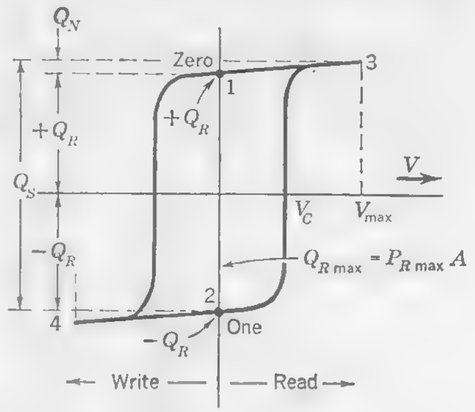


FIG. 12.96. P-E loop of c-domain barium titanate.

Individual storage cells as well as a multicondenser structure on a single crystal slab can be produced.

Production of barium titanate storage cells involves the following procedures:

1. Crystalline plates are sorted according to thickness.

2. Plates are cut into desired sizes and processed to a desired thickness by etching them down. For etching, hot (150°C) phosphoric acid may be used.

3. The plates thus prepared still possess a large amount of a domains and have to be converted into c-domain crystals by field orientation. For this purpose the crystals are placed between the electrodes of a condenser having a somewhat larger air gap than the thickness of the crystal plates; the excess gap volume is filled with glycerin, which is slightly conductive. Using a battery, a field of about 1,000 volts per cm is produced within the gap of the condenser; the condenser is then heated beyond the Curie temperature while the field is alternated a number of times. After slow cooling, the crystal plates are converted into c-domain plates.

4. The crystals thus prepared are then cleaned and washed using hydrochloric acid and distilled water.

5. Finally, electrodes of platinum, gold, or silver are usually vacuum-vapor-plated through a mask onto the crystal surfaces.

6. These units are then placed in a socket, usually by a cold-soldering process using silver paste, and the whole is coated with a protective lacquer.

Figure 12.97 shows a single storage cell on a slab of ferroelectric crystal plate.

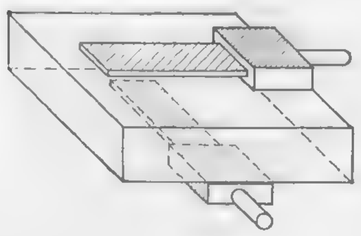


FIG. 12.97. A storage cell.

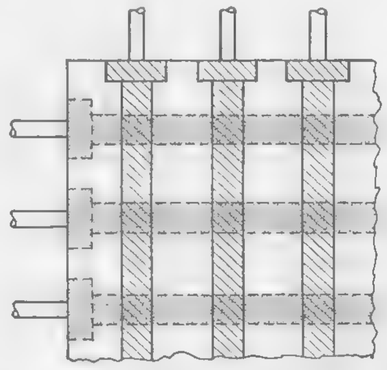


FIG. 12.98. An $n \times n$ array of storage cells.

Figure 12.98 shows a section of an $n \times n$ ferroelectric matrix; columns and rows form a crossbarlike structure.

Figure 12.99 presents five twin storage cells on a slab, useful in a nondestructive storage structure.

12.8.3. The Bistable Condenser as a Circuit Element. Figure 12.100 shows the basic circuit of a single ferroelectric memory flip-flop. C_s denotes the storage cell, and the load, which may take various forms, is symbolized by the box. In order to store digital information in the storage cell, a reference frame must be chosen. The $+Q_r$ point on the hysteresis loop is the normal rest state of the element and it corresponds to a binary 0. A negative voltage of sufficient amplitude will switch the element from the $+Q_r$ state to the $-Q_r$ state; let the $-Q_r$ state correspond to a binary 1. Thus the memory elements of an array of condensers normally store binary 0's; digital words are written into the array by the application of negative write pulses corresponding to the digital 1's present in the word.

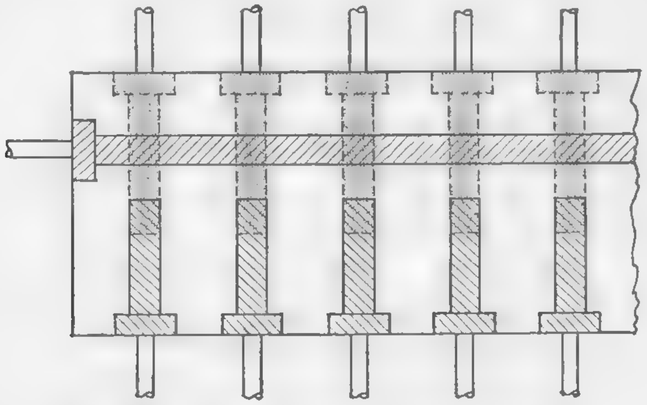


FIG. 12.99. Basic ferroelectric storage cells.

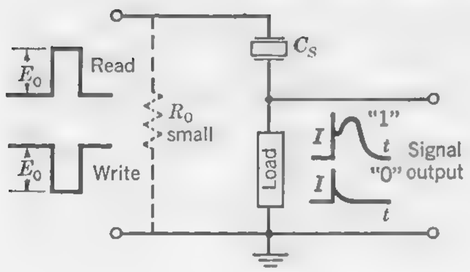


FIG. 12.100. Basic storage circuit.

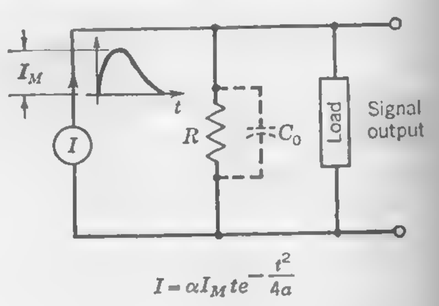


FIG. 12.101. Equivalent circuit.

If condenser C_s stores a binary 1, the sensing or read-out pulse causes a switching transient in the load circuit and a large signal will appear on the output terminal indicating the binary 1 state of the selected condenser C_s; if the storage condenser stores a binary 0, no switching will occur and only a small elastic transient will be caused. The ratio of charges represented by the switched and nonswitched transient is called the "signal-to-noise" ratio of the device which, of course, is dependent on the amplitude and duration of the applied voltage pulse.

The switching transient.

Referring to Fig. 12.101, the current due to the switched charge can be approximated by

$$i(t) = \alpha I_m t e^{\frac{-t^2}{4a}} \tag{12.8.1}$$

where the maximum switching current I_m and the parameter a depend on the applied switching field, and where α, a constant, depends on P_r and the area of the ferroelectric element. Typical values of these parameters for very small storage condensers with an area of 8×10^{-4} cm^2 and an applied field of 6,000 volts per cm are $a = 0.5 \times 10^{-12}$, $\alpha = 1.6 \times 10^{6}$ sec^{-1}, and $I_m = 60 \times 10^{-3}$ amp.

Switching Resistance R. Figure 12.102 illustrates the variation of switching transient when switching pulse amplitude is varied. It shows that I_{max} increases with increasing pulse amplitude. Plotting I_{max} against pulse amplitude, the thinner nonlinear curve shown in Fig. 12.103 is obtained. Switching can be broken down into

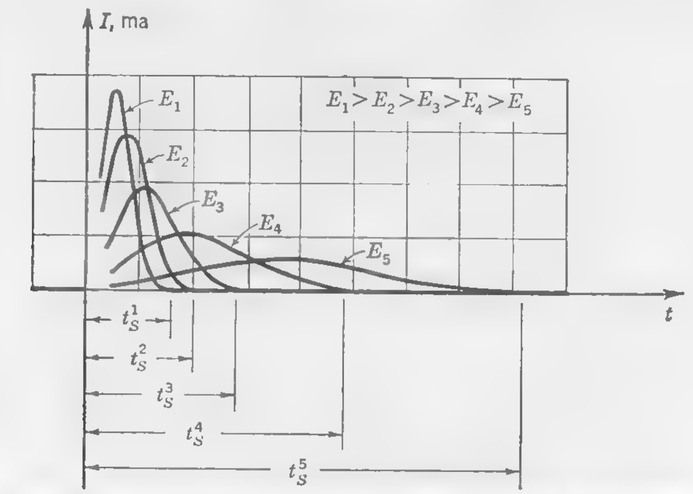


FIG. 12.102. Switching current for different pulse amplitudes.

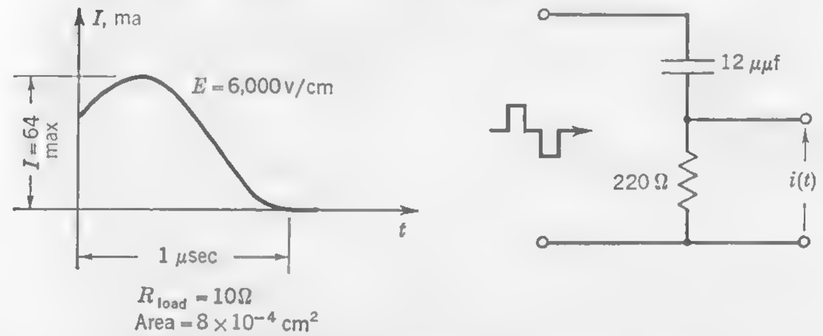


FIG. 12.103. Transient response with resistive load.

two processes (*cf.* Merz[51]): nucleation of domain spikes and growth of domain wedges. Correspondingly, the low-field-strength portion connected with the process of domain spike nucleation is described by an exponential relationship

$$I_{max} \cong e^{\frac{-\alpha(T)}{E}} \tag{12.8.2}$$

and will be referred to as the current-operated region. $\alpha(T)$ is a parameter representing some sort of activation field which is a function of temperature. In the high-field region i_{max} is practically a linear function of applied field

$$i_{max} \cong \mu(T)(E - E_c') \tag{12.8.3}$$

and will be referred to as the voltage-operated region. μ is the mobility (approximately 2.5 cm^2 per volt sec at room temperature) while E_c' is a constant corresponding to a kind of coercive field which in ferroelectrics is due to the exponential current-operated region and is not a well-defined material property. In this linear voltage-operated region nucleation, as well as growth velocity of domains, is extremely rapid compared with the exponential current-operated region and the crystal behaves similarly to an ohmic resistance R.

The reciprocal slope of this curve represents the internal resistance of a ferroelectric storage cell during switching and is regarded as the switching resistance of a storage cell. If the device switching resistance is of interest, a very low resistive load (about 5 to 10 ohms) is used and the transients obtained represent the current produced by the storage condenser as a generator.

If similar plots are taken with increasing load resistance, a family of switching curves can be obtained which are characteristic of a particular storage cell and, therefore, such plots are called *dynamic-switching-resistance curves*. Once this family of curves is measured, an equivalent circuit as shown in Fig. 12.101 can be established and the electrical values of the equivalent circuit can be determined. The equivalent circuit of a bistable storage cell can thus be regarded as a current generator with resistance in parallel which is determined only by the properties of the storage cell such as the coercive field, P_r, area, and thickness of the crystal cell, provided that the internal resistance of the switching pulse source R_0 is small. With the equivalent circuit and the equation of the transient, circuit analysis can be carried out.

Load Impedance and Circuits. Different types of impedance may be applied as a load such as a resistance, capacitance, inductance, or a nonlinear resistance of any type. Examples of such loads are resistors, capacitors, pulse transformers, and diodes or transistors. Load circuits may be formed by combining different impedances as desired. The character of the output signal depends on the type of load chosen (Figs. 12.102, 12.103, and 12.104). The load impedance may be a part of the read-out circuit.

FIG. 12.104. Transient response with an *RC* load.

Switching Time t_s. t_s is the time required to reverse a major fraction (say 90 percent) of all the electric dipoles. Referring to Fig. 12.102, it appears clearly that t_s decreases with increasing pulse amplitude. The switching time is described by the following equation:

$$t_s \cong \beta \frac{d}{E - E_c'} \qquad (12.8.4)$$

where d is the thickness of the crystal plate, E is the applied switching field, E_c' is a kind of coercive-field strength as before in Eq. (12.8.3), and β is a constant. It is noted that, although t_s can be kept very low (between 0.1 and approximately 2 μsec), the repetition rate of a storage cell is limited because of the heat generated by internal friction during switching (t_s is also sensitive to crystal imperfections). At present such condensers can be operated with a repetition rate in the kilocycle region having electrodes 16×10^{-6} sq in. in area. As the electrode area increases the upper frequency limit decreases. Large storage cells with about 400×10^{-6} sq in. have a safe upper frequency limit of about 10 kc. The use of heat sinks as well as further development in storage-cell production may raise these limits considerably.

12.8.4. Transpolarizer.[64] The transpolarizer, analogous to its magnetic counterpart the transfluxor, consists of at least two condensers with a preferable crystalline ferroelectric dielectric and a nearly rectangular hysteresis loop. Control of polarization transfer through two or more applied ferroelectric dielectric paths in series represents a means for storing and gating electrical signals and, in general, means to control circuit impedance in any predetermined manner according to a stored setting.

The impedance of a circuit can be set to any predetermined level by a single pulse and it will remain for an indefinitely long time. Transmission of an a-c power can therefore be controlled or set to a desired level by a single setting pulse. This level may assume any value between zero and the maximum limits of the device and perform either an "on-off" or "continuous" stored control.

The operation of a two-path transpolarizer is described below. Figure 12.105 represents a two-path transpolarizer comprising two identical ferroelectric storage cells

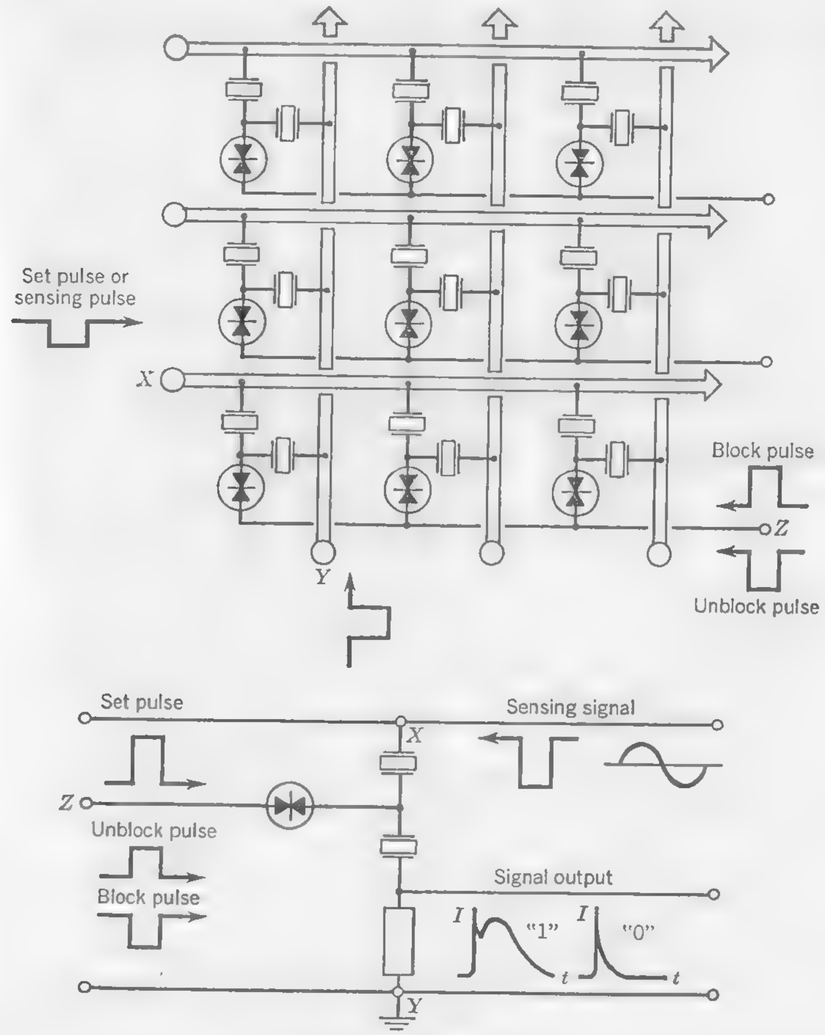
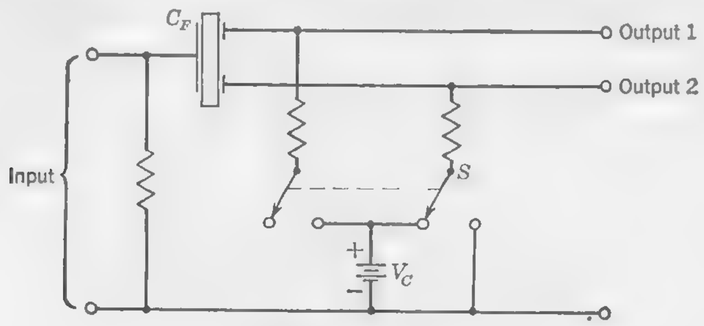


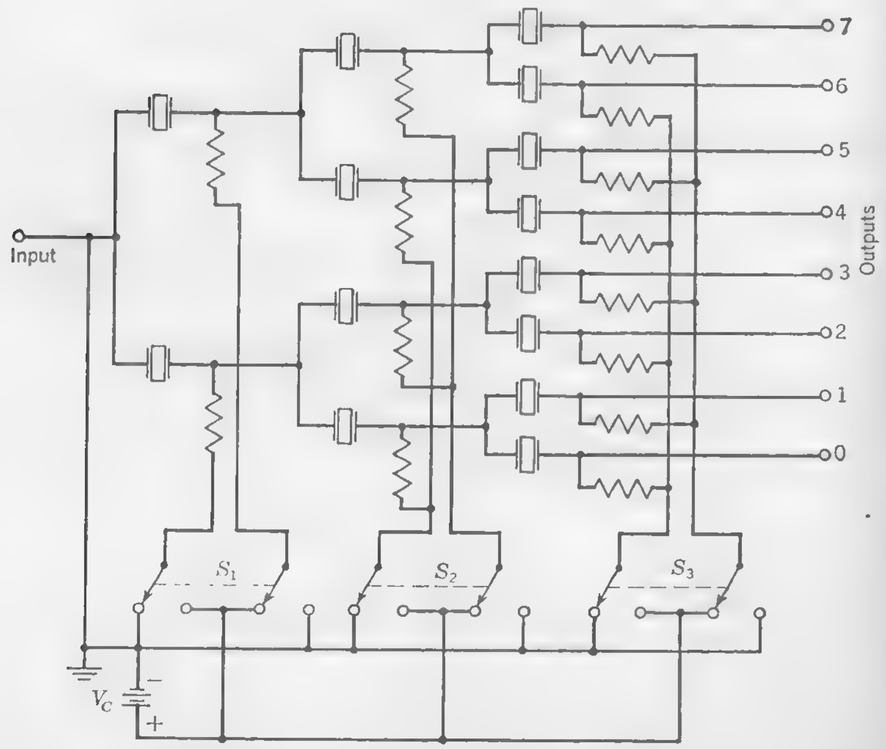
FIG. 12.105. Transpolarizer.

connected in series. When the polarization of both crystals is identical a sensing pulse opposing these polarization directions can switch both storage cells, or if a sine wave is applied as a sensing signal it will find a low impedance since each half period can switch both condensers. If, however, the polarization of the two crystals is opposite for either polarity of a sensing pulse or sine wave, one of the two storage cells will be driven increasingly in saturation where dielectric constant is small and therefore represents a small capacity, i.e., high impedance, and neither of the storage cells can be switched. When both cells are polarized in the same direction a binary 1 is stored and when the polarization in the two cells is opposite a binary 0 is stored. The set and block or unblock pulses are supplied through double anode saturation diodes as shown in Fig. 12.105 in order to isolate these pulse sources from the circuit itself.

12.8.5. Ferroelectric Switches. Earlier developments such as the switch of Dudley Buck use the series and parallel combination of biased and unbiased ferroelectric capacitors. Those unbiased can be switched by a voltage pulse or an a-c voltage; those biased beyond the coercive voltage are in saturation where dielectric constant is small and impedance is high and, for this reason, block the path of a pulse or an a-c



(*a*) Two-position switch



(*b*) Eight-position switch

FIG. 12.106. Ferroelectric-switch schematic diagram.

voltage. This principle is shown in Fig. 12.106*a* while Fig. 12.106*b* shows an eight-position ferroelectric switch. A voltage pulse or an a-c voltage at the input terminal is distributed to one of the eight output terminals as dictated by the voltages on the control wires. Control-wire pairs are driven by binary devices, here represented by toggle switches S_1, S_2, and S_3 in such a way that one wire of each pair is grounded and the other is held at a voltage V which exceeds the coercive voltage of the ferroelectric capacitors. With the switches set in a given pattern all paths except one have at least

one capacitor biased in a saturated region where the polarization cannot change. The single path whose three capacitors are unbiased is the selected path. When the input terminal is excited all three capacitors in this path traverse their hysteresis loops, thus allowing a current to flow between input and selected output.

The transpolarizer, as shown in Fig. 12.105, enables one to produce similar tree-switch configurations with the advantage that no bias is needed; blocking or unblocking of a particular path needs but a single pulse distributed to the proper control leads of the switch.

12.8.6. A Shift Register.[4] Figure 12.107 shows a shift register having completely independent parallel or series inputs and outputs or any combination of these. Practical operating speeds may be from 0 to above 10,000 pps, although the ferroelectric

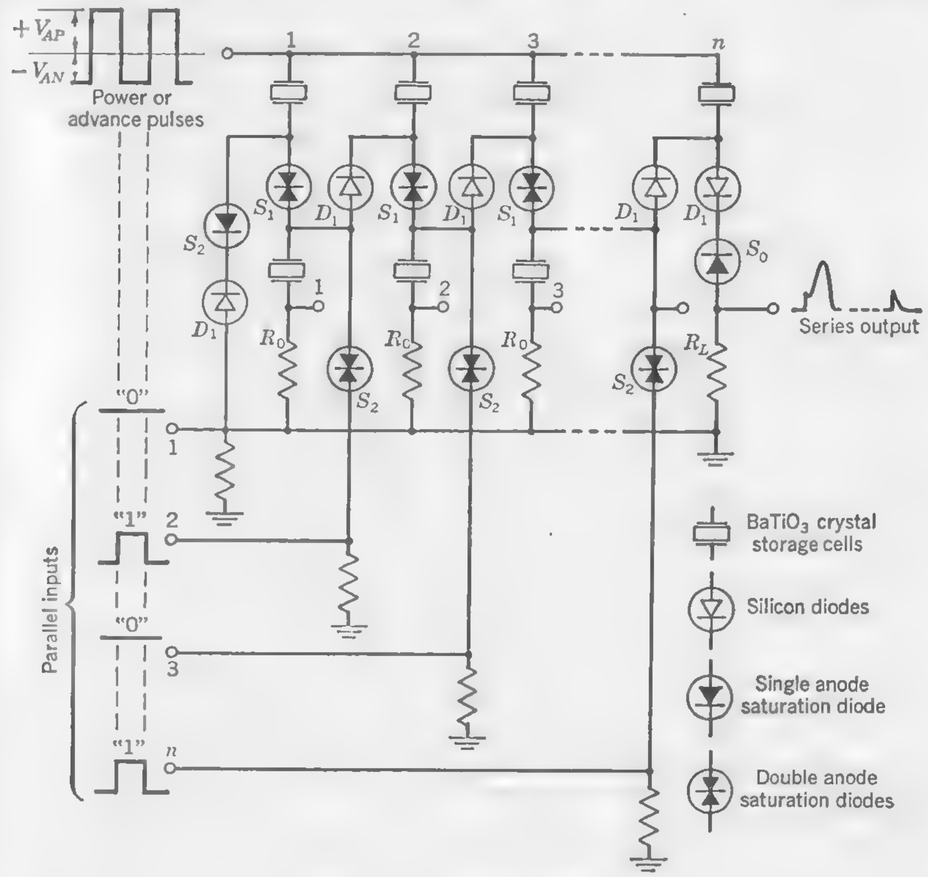


FIG. 12.107. Shift register.

shift register is probably most useful at speeds below 2,000 pps when very long output pulses are needed. The small size of the ferroelectric units, the low power consumption at low speeds, and completely transistorized drive circuits make the ferroelectric shift register attractive for many applications.

The operation of the shift register is explained by the circuit diagram shown in Fig. 12.107. This diagram illustrates an n-stage shift register with connecting links properly made between the stages for each step in the operation. When the shift register is empty, the crystals will be poled in opposite pairs. Positive and negative advance pulses applied to the common power bus will not be able to switch any of the crystals, and the shift register will remain in this state until storage takes place.

Binary 1's are stored or set in the shift register by applying a positive input pulse to selected upper crystals during the negative power pulse through inputs 1, 2,

3, . . . , n. The negative power pulse and positive input pulses as shown on input leads 2 and n add themselves and switch through diodes D_1 and saturation diodes S_2, the selected upper crystals 2 and n, so that these are then poled with their corresponding lower crystals in aligned pairs and a positive power pulse is next applied. This positive power or advancing pulse exceeds the amplitude of the negative pulses by the saturation voltage of the S_1 diodes in order to overcome this voltage drop which always subtracts from the positive voltage applied across the crystals and switches the selected upper crystals 2 and n back in their original state, causing an output signal from n and reversing the polarity of the lower crystal in section 2. A negative power or advancing pulse is next applied across the crystals through D_1 diodes which will cause the lower and upper crystal units of stages 2 and 3 to switch in series. Section 2 returns to its 0 state and section 3 is left in a stored binary 1 state and one shift cycle is completed. The negative advance pulse acting through D_1 diodes appears nearly in its full negative power voltage across the crystal because the D_1 diodes are conducting in the forward direction. The next positive power pulse and negative advance pulse would shift the stored binary 1 from section 3 into the nth section, and so on. Alternate negative and positive pulses simply shift a stored 1 from any section into the output load, leaving the register finally empty. Figure 12.108 shows the series output pulses from a 10-stage parallel-input ferroelectric shift register in which 290 × 10^{-6} sq in. electrode units have been used and for which the pulse rate is 1,000 pps.

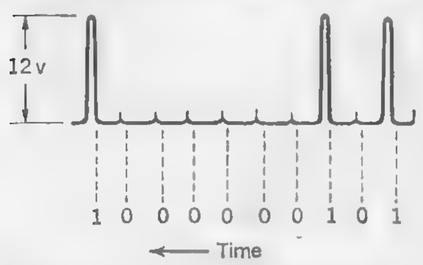


FIG. 12.108. Output pulses from a shift register.

The above description has covered the case of parallel inputs and series outputs. The shift register would have been filled just as easily in series by setting or not setting the upper crystal of stage 1 on successive negative pulses. In addition, parallel outputs could have been taken by placing load resistors between ground and the lower electrode of each lower crystal so that full conversion between a series input and parallel outputs, or parallel inputs and a series output, is possible.

The shift register is also capable of having any of its parallel outputs combined with each other without affecting series shifting down the register.

12.9. DIODE-CAPACITOR MEMORY SYSTEM*

12.9.1. Introduction. From several points of view a capacitor is an ideal storage element for a computer, the most important being that it stores charge directly without the necessity of converting the energy to some other physical form. This direct use of charge greatly simplifies and speeds up any system. In addition, a capacitor is stable, cheap, and rugged, and its parasitic losses are extremely low.

Unfortunately, in order to store charge on a capacitor for a long time, it is necessary to have some form of switch, and switches become the real problem. Various systems involving mechanical switches have been suggested—such as rotary electrostatic drums—but these suffer from the usual mechanical difficulties and still do not exploit the high-speed possibilities of condenser storage.

There are several forms of high-speed electronic switches which may be used with capacitors to form a storage system. Gas tubes, vacuum tubes, and semiconductor devices fall in this category, but for large-scale high-speed memory, only the semiconductor diode seems practical at this time. The cold-cathode gas diode may be a possibility for the future.

This discussion will be limited, therefore, to the type of memory called the diode-capacitor, or Dicap, system, which was suggested and developed at the National Bureau of Standards.

The following characteristics of a Dicap memory are set down here only as a qualitative guide to the reader. The primary advantage is speed, and at the present

* By Arthur W. Holt.

time this type of memory provides the fastest access available in a practical form. Using silicon-junction diodes the speed is limited by the reverse transient response of the diodes, but several random accesses per microsecond may nevertheless be obtained. Using point-contact diodes or vacuum diodes, the access rate can be increased still further, limited only by the speed of vacuum-tube circuitry necessary to drive the memory. The primary disadvantage (at the time of writing) is the high cost of diodes. Two diodes per bit are required. Somewhat offsetting this expense is the inexpensiveness of the access circuitry and the reading amplifiers. Other advantages are: all signals are high level and low impedance, all components are light and can take heavy physical abuse, and all components are in large-scale production.

This type of memory has not yet been used in any large-scale application, primarily because the demand for high speed has not been great enough to warrant the expense. The scheme was first proposed in 1951, and several small-scale models were tested experimentally. In 1953, a medium-sized experimental model was built capable of storing 256 words of 8 bits each. Each random access took 1.75 μsec of active time, although the system was operated on a 6-μsec cycle. This model was tested extensively using the SEAC computer at the National Bureau of Standards, and it proved to be thoroughly reliable. Several other laboratories have experimented with the system, and a small-scale version has appeared in at least one commercial computer.

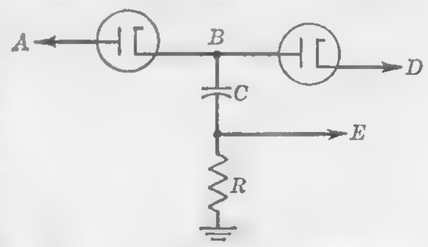


FIG. 12.109. Basic storage-element diode-capacitor memory.

12.9.2. Basic Cell. The basic storage element of this memory is shown in Fig. 12.109. The connection E is used for both reading and writing, while the two diodes between A and D are used as a "squeezer" to connect the capacitor to the reading-writing circuits, as will be seen. During holding, both diodes are biased in their back direction. For example, A might be held at −4 volt with respect to ground, and D held at +4 volts. Then if the capacitor has a charge of, say, 2 volts, both diodes will be biased in their back direction and only small currents will flow into or out of the capacitor. Now for reading, suppose that both points A and D are forced to ground potential ("squeezed"). This will cause one or the other diode to conduct and a voltage will appear across the resistor R. If C is charged with 2 volts of such polarity as to make its lower terminal (in the figure) more negative than its upper terminal, then when the squeeze occurs there will appear at E a pulse of −2 volt, which then dies out with the time constant RC. This would be recognized by the reading circuits at E as the binary digit 0. If the polarity of the charge on C had been in the opposite direction, the squeeze would have produced a positive pulse instead of negative and would be recognized as the binary digit 1. Thus the content of the storage element has been read, but in doing so it has been (at least partially) discharged and the information lost from the storage element. The information must be rewritten to continue the storage beyond the reading operation.

In order to write (or rewrite) information it is merely necessary to force the lead E to the desired state during the squeeze and hold it there until the squeeze is over. Thus while A and D are at 0 volt, suppose that E is forced to +2 volts and held there at least until A and D are returned to their normal voltages of −4 and +4, respectively. Then the capacitor is left with a charge of 2 volts, and upon the next squeeze it will produce a positive pulse at E. That is, we have written a "1." Obviously the opposite is equally possible: forcing E negative until the end of the squeeze will write a "0." Note that, once A and D have returned to their normal voltages, the charge on the capacitor will be undisturbed by later changes of E, provided the magnitude of E's voltage never exceeds 2 volts. Thus E can have other pulses on it, either positive or negative, and the charge stored on C will remain unaffected because both diodes will remain with backward bias. This is important for organizing many basic storage elements into an efficient memory assembly and is the reason for charging the capacitor to only ±2 volts while biasing the diodes twice as much.

In the description so far the diodes have been implicitly assumed to be ideal, having practically infinite forward conductance and practically zero backward conductance. The effect of finite forward conductance is modest: it will reduce somewhat the output pulse amplitude, and it will determine how long a writing pulse must last to charge the capacitor adequately. The effect of finite backward conductance, however, is critical. During the holding operation relatively long times will elapse, and even minute currents through the diodes will disturb the capacitor charge. The unit would gradually leak toward a condition of no charge on the capacitor, or even a condition in which the sign of the charge is reversed. Thus the permissible duration of the holding operation is determined by the rate at which the capacitor charge leaks through the diodes' back current. Arbitrarily long storage of information is achieved through regeneration: before the capacitor charge can change to a point where there is danger of losing the information, the memory-control circuits routinely read the content of each cell and rewrite it accordingly.

What is needed at point E, then, is an amplifier which will sense the polarity of I during the early part of the squeeze period, together with a gate structure which will force E to the desired polarity during the latter part of the squeeze period. For reading or regeneration, E is forced to the same polarity that was read; for writing new information the polarity to which E is forced is independent of what was read but is determined by the new information being written. Such a gating amplifier is easy to construct. The amplification required is very moderate, since its input is a pulse whose amplitude is of the order of 1 volt. The gating can be accomplished with standard techniques.

12.9.3. Arrangement of Cells in Matrix. In order to achieve acceptable efficiency it is essential that one such gating amplifier serve many basic storage elements.

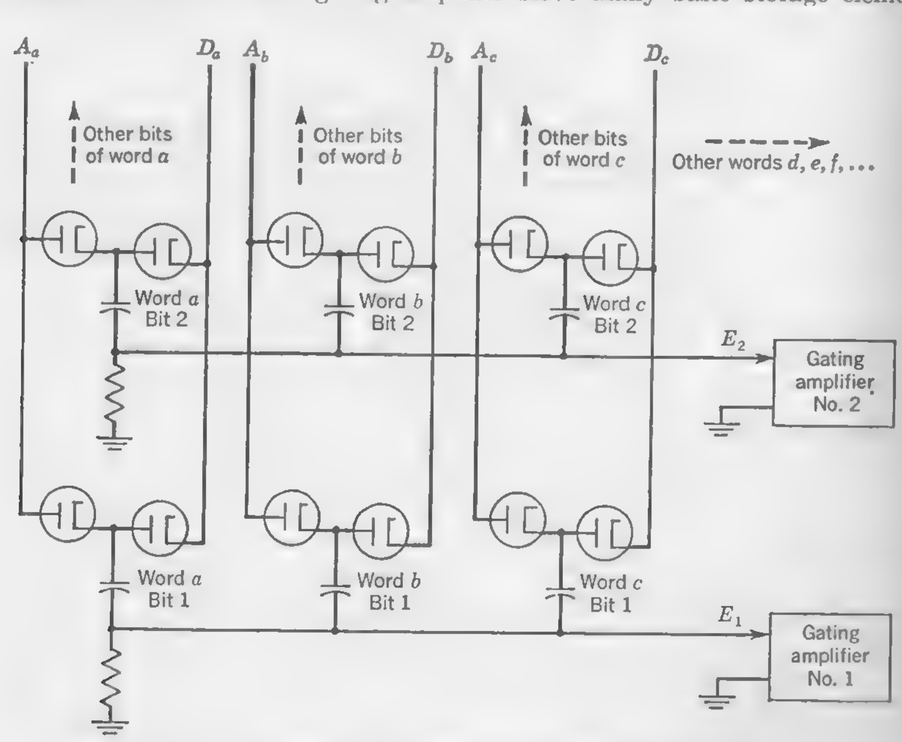


FIG. 12.110. Assembly-of-words diode-capacitor memory.

Figure 12.110 shows how this is done. The buses A and D are made common to all the bits of a particular computer word, and a particular gating amplifier serves the same bit on each of many words. Thus for 256 words of 45 bits each there would be 256 pairs of leads A and D, and 45 gating amplifiers. For reference to word b

the buses A_b and D_b would be squeezed to zero voltage, while all the other pairs would be held at their normal values of -4 and $+4$ volts. In this way each gating amplifier receives a pulse from its bit of the selected word; so the word is available in parallel at the gating amplifiers. These amplifiers can then write into this word, or rewrite it, without affecting the other words, since all diodes in the other words remain with backward bias as already described. After the squeezing buses on word b are returned to normal, any other word may be referred to in the same way. This gives a fully parallel random-access memory. Regeneration is of course handled by having the memory control intersperse regeneration cycles between the computer access cycles. For the regeneration cycles, the words are read one after the other and rewritten to their former state.

12.9.4. Access Circuitry—The Transformer-diode Coincidence Gate. The system described so far achieves reasonable efficiency for the gating amplifiers but requires a selection circuit capable of squeezing the appropriate pair of buses for a particular word. This could be accomplished by a diode matrix, but the customary form of such a matrix has large standby currents. In this memory the squeezing buses require relatively large currents; the resultant selection matrix is feasible but draws large amounts of standby power. To avoid this, a selection matrix using transformers and diodes is used as shown in Fig. 12.111. This gives a matrix which has no standby

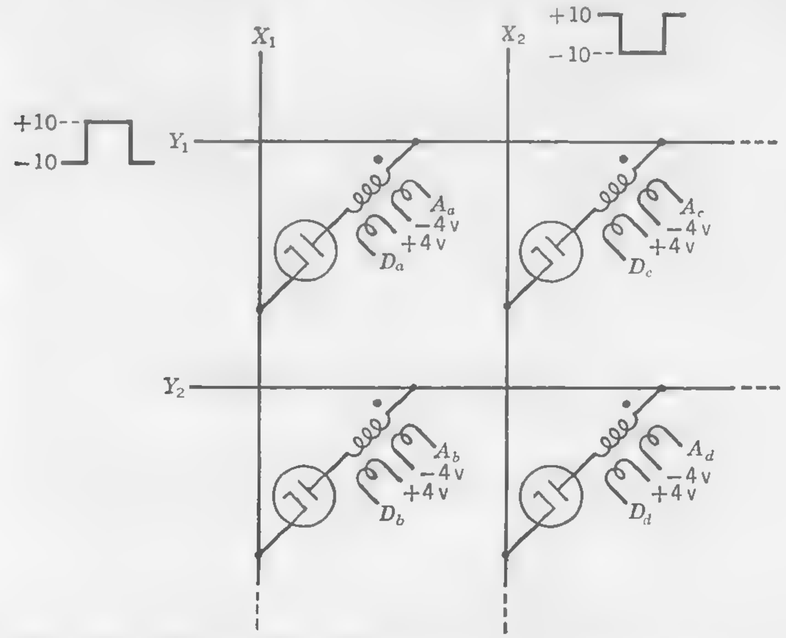


FIG. 12.111. Transformer-selection-matrix diode-capacitor memory.

power requirement, although it does require more input drivers than would be necessary with a multidimensional diode matrix. For the transformer-diode matrix, $2n$ inputs are required to select from among n^2 words. The matrix is made up of two sets of crossing buses (X and Y in Fig. 12.111). At each crossing a diode and transformer are connected as shown. Normally all the X buses are held at, say, +10 volts, and all the Y buses are held at -10 volt. This puts backward bias on the diodes associated with each transformer; so no current flows through any transformer. If one X bus is dropped to -10 volt, still no current will flow; but if simultaneously one of the Y buses is raised to $+10$ volts, then just the one transformer at the crossing of these two buses will receive a signal. Thus if, in the figure, X_2 is lowered to -10 volt and Y_1 raised to $+10$ volts, the transformer secondary connected to buses A_c and D_c will squeeze the voltage on these two buses together. This will select the desired word.

12.9.5. Typical Silicon Diode Specification. Since the silicon-junction diode has, at this time, characteristics which are far superior to any other diode for the Dicap application, its characteristics will be used as illustrations for the various points in this discussion. The typical specification gives an approximate idea of both what is available and what are reasonable limits for Dicap application. Although the specifications change somewhat with temperature, the design can generally take these into account with ease. A design can be made for quite extreme temperature ranges, if necessary.

A Typical Silicon-junction Diode Specification (at 25°C)

	For Dicap application	Comments
Reverse current	Max: 10^{-9} amp at 5 volts Min: 10^{-11} amp at 5 volts	Want this current as small as possible
Reverse breakdown voltage	The high-production units have a minimum of about 17 volts, which is suitable	
Forward dynamic resistance	Max: 75 ohms at 1.5 volts Min: 25 ohms at 1.5 volts	Want this as low as possible, but fairly uniform
Forward step	Max: 0.75 volt Min: 0.65 volt	Want uniformity here. No problem in production
Reverse capacitance	Max: 20 $\mu\mu$f at 4 volts Avg: 15 $\mu\mu$f Min: 2 $\mu\mu$f	Want this as low as possible. Designer must know average value, but individual values are not important
Delta at 1 Mc (ratio of reverse to forward charge)	Max: 0.3 Min: 0.1	Want this as low as possible. Individual value is important

12.9.6. Holding Equations. For *holding*, the circuit and equivalent circuits are given in Figs. 12.112 and 12.113. The *holding* equations are given in Table 12.32.

Table 12.32. Holding-time Equations

Let V_c be the voltage across the capacitor at time t
Let V_0 be the initial voltage at $t = 0$

$$V_c = V_0 - \frac{R_1R_2}{R_1 + R_2}\left(I_1 + \frac{V_0 - V_1}{R_1} - I_2 + \frac{V_0 - V_2}{R_2}\right)\left[1 - \exp - \left(\frac{R_1 + R_2}{R_1R_2C}\,t\right)\right] \text{ volts} \quad (12.9.1)$$

The time t for capacitor to discharge from V_0 to V_c is

$$t = \frac{R_1R_2C}{R_1 + R_2} \ln \left[\frac{\frac{R_1R_2}{R_1 + R_2}\left(I_1 + \frac{V_0 - V_1}{R_1} - I_2 + \frac{V_0 - V_2}{R_2}\right)}{V_c - V_0 + \frac{R_1R_2}{R_1 + R_2}\left(I_1 + \frac{V_0 - V_1}{R_1} - I_2 + \frac{V_0 - V_2}{R_2}\right)}\right] \text{ sec} \quad (12.9.2)$$

Let $V_1 = 0$ $\quad V_{\text{final}} = V_F = \left(\frac{R_1}{R_1 + R_2}\right)V_2$ and $R_p = \frac{R_1R_2}{R_1 + R_2}$

Then

$$V_c = V_0 - (V_0 - V_f)(1 - e^{-t/R_pC}) \quad (12.9.3)$$

$$t = +R_pC \ln\left(\frac{V_0 - V_f}{V_c - V_f}\right) \quad (12.9.4)$$

Equations (12.9.1) and (12.9.2) are based on the assumption that the diode characteristics can be approximated by a constant current in parallel with a constant resistance, and this is a very good approximation over the few volts excursion used in Dicap memory work. It is interesting to note that, if two diodes with equal absolutely constant current characteristics are used to form a storage cell, this cell will permanently retain whatever storage level is set into it.

Equations (12.9.3) and (12.9.4) are accurate if the asymptotic current can be neglected. These equations give much better visualization of the effects taking place, and they are very useful in making quick checks.

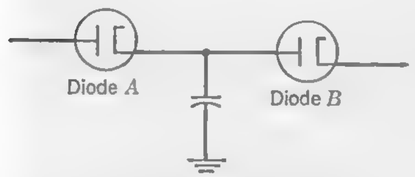


FIG. 12.112. Bit circuit.

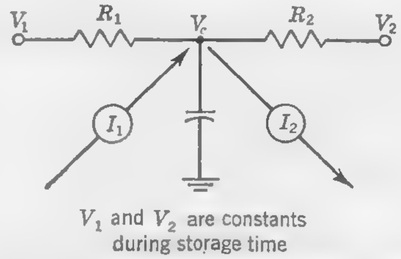


FIG. 12.113. Equivalent bit circuit during holding time.

Example of Calculation of Holding Period. Assume two diodes of the following characteristics:

	Diode *A*	Diode *B*
Reverse current at 10 volts	$I_{R1} = 40 \times 10^{-9}$ amp	$I_{R2} = 3 \times 10^{-9}$ amp
Extrapolated zero currents	$I_1 = 5 \times 10^{-9}$	$I_2 = 1 \times 10^{-9}$

We compute the constant resistances from the reverse currents at 10 volts.

$$R_1 = \frac{10}{40 \times 10^{-9}} = 2.5 \times 10^8 \text{ ohms}$$

$$R_2 = \frac{10}{3 \times 10^{-9}} = 3 \times 10^9 \text{ ohms}$$

Using Eq. (12.9.2),

Dot case (point *B* storing positive level)

Let $V_0 - V_1 = 6$ volts
$V_0 - V_2 = -2$ volts
$V_c - V_0 = -1$ volt

This last condition requires that not more than 1 volt be lost during storage time. Result:

$$\tau_{\text{dot}} = (39 \times 10^8)C \text{ sec}$$

Dash case (point *B* storing negative level)

Let $V_0 - V_1 = 2$ volts
$V_0 - V_2 = -6$ volts
$V_c - V_0 = +1$ volt

Result:

$$\tau_{\text{dash}} = -(0.83 \times 10^8)C \text{ sec}$$

This means that the dash case will always be stable and that the cell "leaks" toward the negative level. Using Eq. (12.9.4) (simplified equation),

Dot case

Let $V_0 = +6$
$V_c = +5$

We calculate

$$V_f = 0.61 \text{ volt}$$

Result:

$$\text{Dot} = (0.46 \times 10^8)C \text{ sec}$$

Dash case

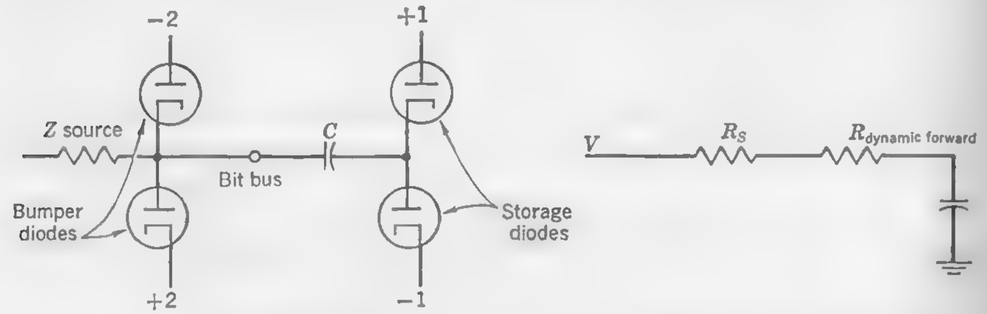
Let $V_0 = +2$
$V_c = +3$
V_f is constant at 0.61 volt

Result:

$$\text{Dash} = -(1.35 \times 10^8)C \text{ sec}$$

12.9.7. Writing Equations. Figure 12.114 shows first the actual circuit *a*, in which the two storage diodes are biased in the forward direction. Note that the "bit bus" must not make excursions greater than a predetermined amplitude during charging; the reason for this, of course, is that the other storage cells on the same "bit bus" may bump into the reverse-bias level and thus lose their charge.

During the charging, all the charging current flows through one storage diode or the other, even though a forward biasing current is used. A suitable equivalent circuit is shown in *b* of Fig. 12.114. For our purposes it is permissible to assume that



(*a*) Typical charging circuit (*b*) Suitable equivalent circuit

FIG. 12.114. Writing circuits.

the driving source is purely resistive, that the forward impedance of the storage diode is purely resistive, and that the effective resistance of the storage diode is its dynamic slope. This latter is an important point which eases the design considerably. The forward biasing of the storage diodes determines this dynamic operating point. Forward transient in silicon-junction diodes can be neglected but may be serious in large-area germanium diodes.

For our purposes, therefore, the charging part of the writing cycle can be expressed by the simple relationship

$$V_{c1} = V\left[1 - \exp\frac{-t}{(R_s + R_{df})C}\right]$$

where V = step function applied at $t = 0$
V_{c1} = voltage across capacitor
R_s = driving-source resistance
R_{df} = dynamic forward resistance of storage diodes

It is clear that the driving-source resistance should be as low as possible for fast writing.

Reverse Transient Response. If silicon-junction diodes are to be used with charging times of less than 1 μsec, the effect of the reverse transient response of diodes must be considered. In Dicap operation, this characteristic simply reduces the amount of charge left on the capacitor.

$$V_{c2} = (1 - \delta)V_{c1}$$

where $\delta = \dfrac{Q_{\text{reverse transient}}}{Q_{\text{forward}}}$

V_{c1} = voltage across C after charging cycle, initial voltage being zero
V_{c2} = final voltage across C after transient

The term $(1 - \delta)$ is very closely related to the "rectification efficiency" sometimes quoted in diode specifications. It is nearly independent of the size of the capacitor used. It is very dependent on the length of time during which the charging occurs—the shorter the time, the larger delta becomes. At present, the value of delta varies considerably, depending on the manufacturer and also on the individual unit. Because of this, it is recommended that the designer measure this characteristic in the laboratory if the desired charging time is to be less than 1 μsec.

As was pointed out, the effect of the reverse transient is primarily in decreasing the output-signal level relative to the original charge. It is still possible to store information successfully, even with charging times as low as 0.1 μsec, but the storage efficiency may be low. If the effect of the reverse transient is too serious, germanium whisker diodes may be used.

12.9.8. Reading Equations. The calculation of the amount of signal delivered to the input of the amplifier is completely straightforward, once the amount of charge remaining on the storage capacitor is known. The only loss in reading is due to the back capacitances of the storage diodes, which are not selected. Figure 12.115 is a

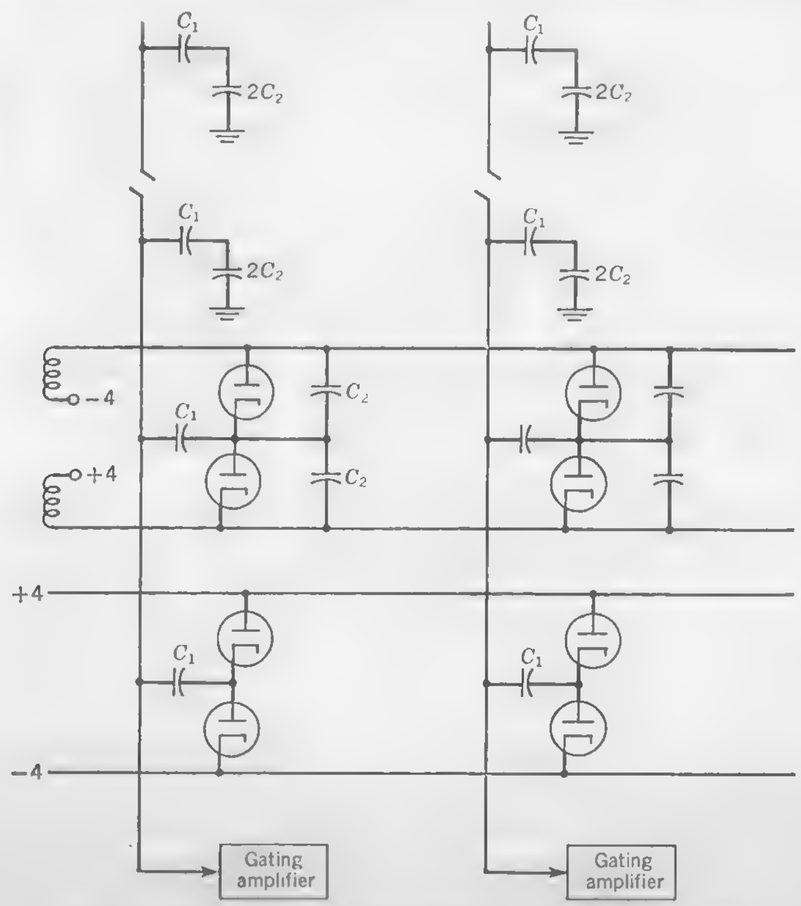


FIG. 12.115. Diagram showing the effect of diode capacitance.

diagram showing this effect. Each diode adds its capacitance C_2 in parallel to ground, and so the selected bit capacitor C_1 has to charge $2NC_2$.

If N = number of bits on one bit bus
C_1 = storage capacitor
C_2 = back capacitance per diode

then, by capacitive division,

$$e_{useful} = \frac{C_1}{C_1 + 2NC_2}(e_{c1})$$

Typically,

If $C_1 = 10{,}000\ \mu\mu f$
$C_2 = 20\mu\mu f$
$N = 500$

$$e_{useful} = \frac{e_{c1}}{3}$$

This is not a true signal loss, however, because the capacitance has shifted the line to a lower impedance. This means that the voltage can be raised by means of a step-up transformer, if desired. In general, no resistive loss need be suffered, since the input impedance of the gating amplifier can be high during the reading time and low during the writing time.

12.9.9. Block Diagram of the System. Figure 12.116 shows the block diagram of the experimental system that was built at the National Bureau of Standards in 1953.

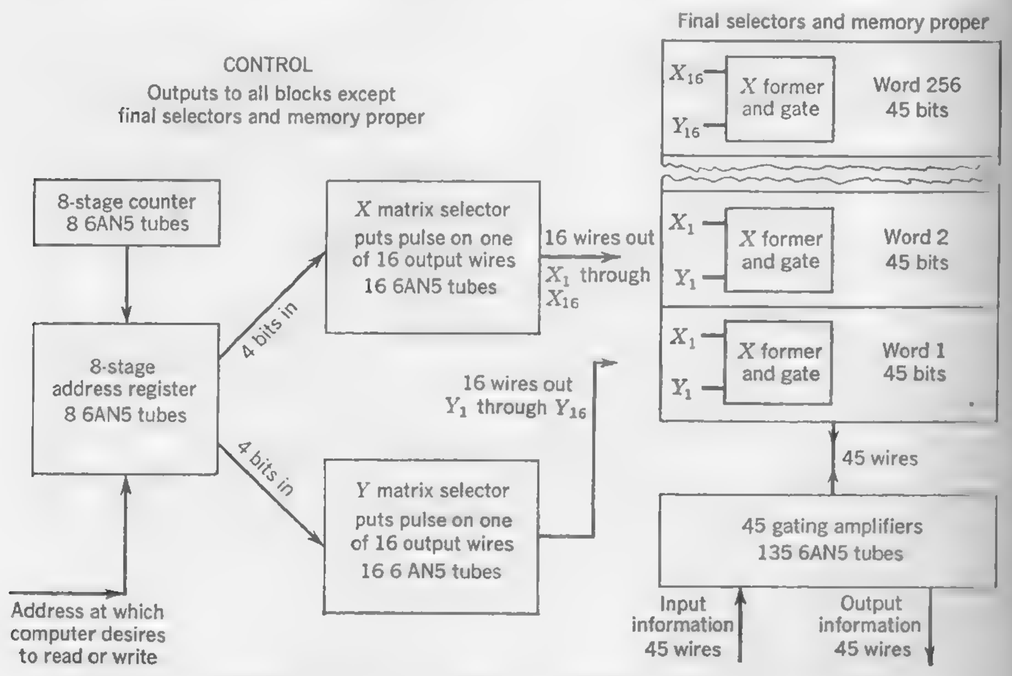

Final selectors and memory proper
CONTROL
Outputs to all blocks except final selectors and memory proper
8-stage counter 8 6AN5 tubes
8-stage address register 8 6AN5 tubes
4 bits in
4 bits in
X matrix selector puts pulse on one of 16 output wires 16 6AN5 tubes
16 wires out
X_1 through X_{16}
Y matrix selector puts pulse on one of 16 output wires 16 6 AN5 tubes
16 wires out Y_1 through Y_{16}
X_{16} Y_{16} X former and gate Word 256 45 bits
X_1 Y_1 X former and gate Word 2 45 bits
X_1 Y_1 X former and gate Word 1 45 bits
45 wires
45 gating amplifiers 135 6AN5 tubes
Address at which computer desires to read or write
Input information 45 wires
Output information 45 wires

FIG. 12.116. Block diagram of a 256-word 45-bits-per-word Dicap memory.

The logical design is similar to the Williams system of electrostatic storage in that regeneration must be intermixed with information access. This is accomplished by feeding the address register from either the regeneration counter or from the computer address bus.

12.9.10. Access Drivers. Figure 12.117 shows a typical stage of the Y matrix selector. The X matrix stage differs from it only in that the polarity of the secondary is reversed and the secondary bias voltage is $+10$ volts instead of -10.[50]

12.9.11. Gating Amplifiers. Figure 12.118 shows the block and logical diagram for a typical gating amplifier. The gating shown is based on positive logic and is common to nearly all kinds of memory systems. The box labeled "Write Driver" merely gives out a positive pulse on the "plus out" line if the input is positive, or a

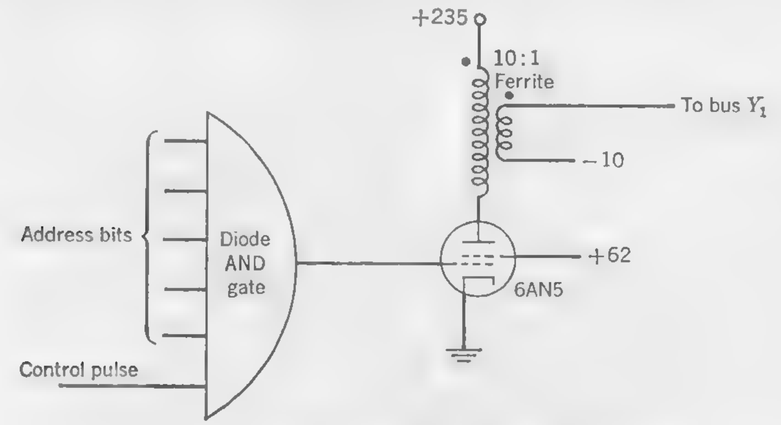


FIG. 12.117. Typical stage of the Y matrix selector.

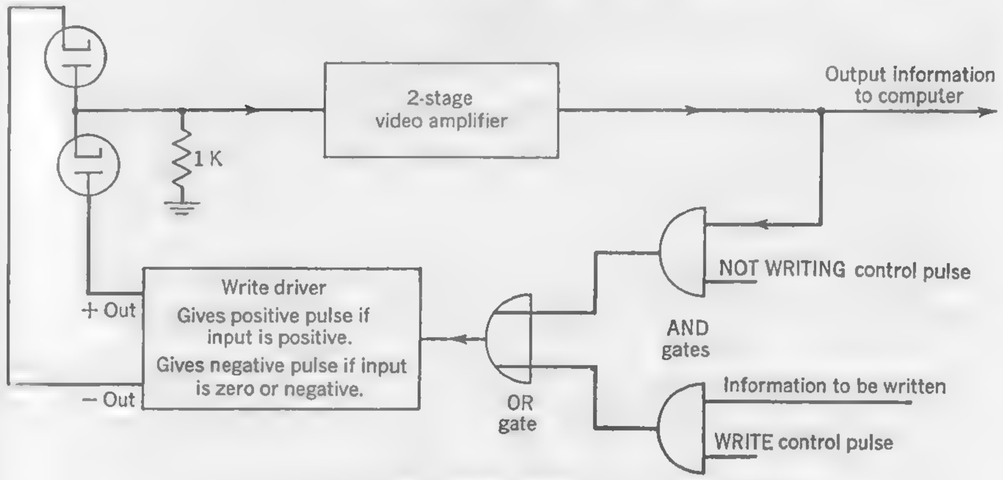


FIG. 12.118. Block and logical diagram of gating amplifier.

negative pulse on the "minus out" line if the input is zero or negative. It is a function which can be designed in any number of different ways, using diode gating and perhaps one vacuum tube. The whole gating amplifier can be designed with three or four vacuum tubes and perhaps 20 semiconductor diodes.

12.9.12. Checking and Maintenance. An interesting form of self-checking can be obtained by taking advantage of the fact that the input is either positive or negative. The gating can be designed to recognize three levels: acceptably positive, acceptably negative, and unacceptable. This would give an early indication of a marginal component.

In addition to the usual methods of marginal checking, most of which are based on power-supply variations, it is possible with Dicap to utilize "regeneration slow-down." This merely means that the length of time between each regeneration is increased—cells that fail too early can be replaced. The control equipment necessary to accomplish this consists only of a few extra flip-flops.

12.9.13. Physical Construction. The physical construction of a Dicap memory can be kept very simple and neat. One arrangement in particular is worthy of note. An etched circuit board, 21 by 10 in., contains all the components for 8 words of 45 bits

each, including the "transformer AND gates." All components are soldered in place. This is mounted as a drawer which slides into a cabinet. Fifty-two heavy spring fingers, which are riveted to the various buses of the etched board, make contact with matching buses on the back of the cabinet. Gating amplifiers and access circuits can be housed at the center of the cabinet with similar construction if so desired.

References

1. Allard, L. S., and R. T. Hill: Switch and Storage Tubes, *Wireless Eng.*, **28**:187–191 (June, 1951).
2. Anderson, J. R.: Ferroelectric Materials as Storage Elements for Digital Computers and Switching Systems, *J. AIEE*, **71** (10): 916–922 (October, 1952).
3. Anderson, J. R., G. W. Brady, W. J. Merz, and J. P. Remeika: Effects of Ambient Atmosphere on the Stability of Barium Titanate, *J. Appl. Phys.*, **26** (11): 1387–1388 (November, 1955).
4. Anderson, J. R.: A New Type of Ferroelectric Shift Register, *IRE Trans. on Electronic Computers*, **EC-5** (4): 184–191 (December, 1956).
5. Anderson, J. R.: A New Type of Ferroelectric Shift Register, *Proc.* 1956 *Electronic Components Symposium*, Washington, D.C.: 219–226, 1956.
6. Bartik, W. J., and T. H. Bonn: A Small Coincident-current Magnetic Memory, *IRE Trans. on Electronic Computers*, **EC-5** (2): 73–78 (June, 1956).
7. Baumann, D. M.: A High-scanning-rate Storage Device for Computer Applications, *J. Assoc. Computing Machinery*, **5** (1): 76–88 (January, 1958).
8. Bay, Z., and N. T. Grisamore: High-speed Flip-flops for the Millimicrosecond Region, *IRE Trans. on Electronic Computers*, **EC-5** (3): 121–125 (September, 1956).
9. Bay, Z., and N. T. Grisamore: Pulse Generator and High-speed Memory Circuit, *IRE Trans. on Electronic Computers*, **EC-5** (4): 213–218 (December, 1956).
10. Begun, S. J.: *Magnetic Recording*, Murray Hill Books, Inc., New York, 1949.
11. Blachman, N. M.: On the Wiring of Two-dimensional Multiple-coincidence Magnetic Memories, *IRE Trans. on Electronic Computers*, **EC-5** (1): 19–21 (March, 1956).
12. Bobeck, A. H.: *Karnaugh Magnetic Switch*, Document 2541-AHB-KL, Bell Telephone Laboratories, February, 1956.
13. Booth, A. D.: Magnetic Digital Storage System, *Electronic Eng.*, **21**:234–238 (July, 1949).
14. Brower, D. F.: A "One Turn" Magnetic Reading and Recording Head for Computer Use, *IRE Convention Record*, part 4, New York: 95–100, 1955.
15. Busch, G., and P. Scherrer: Eine neue seignette-elektrische Substanz, *Naturwiss.*, **23**:737 (1935).
16. Clark, D. L., and L. L. Merrill: Field Measurements on Magnetic Recording Heads, *Proc. IRE*, **35**:1575–1579 (December, 1947).
17. Cohler, Edmund U.: Transistor Flip-flops for High-speed Digital Computer Applications, *Proc. WESCON Computer Sessions*, Los Angeles: 38–43, Aug. 25–27, 1954.
18. Devonshire, A. F.: Theory of Ferroelectrics, *Advances in Physics*, **3**:85–130 (April, 1954).
19. Eckert, J. P., Jr., H. Lukoff, and G. Smoliar: A Dynamically Regenerated Electrostatic Memory System, *Proc. IRE*, **38**:498–510 (May, 1950).
20. Freeman, James Robert: Pulse Responses of Ferrite Memory Cores, *Proc. WESCON Computer Sessions*, Los Angeles: 50–61, Aug. 25–27, 1954.
21. Fuller, H. W., P. A. Husman, and R. C. Kelner: Techniques for Increasing Storage Density of Magnetic Drum Systems, *Proc. Eastern Joint Computer Conf.*, Philadelphia: 16–21, Dec. 8–10, 1954.
22. Goldsmith, George J.: Ferroelectricity in Colemanite, *Bull. Am. Phys. Soc.*, ser. II, **1** (7): 322 (Nov. 23, 1956).
23. Goodman, G.: Ferroelectric Properties of Lead Metaniobate, *J. Am. Ceram. Soc.*, **36** (11): 368–372 (November, 1953).
24. Guterman, S. S., and W. M. Carey, Jr.: A Transistor-magnetic Core Circuit; A New Device Applied to Digital Computing Techniques, *IRE Convention Record*, part 4, New York: 84–94, 1955.
25. Guterman, S. S., and R. D. Kodis: Magnetic Core Selection Systems, *IRE Convention Record*, part 4, New York: 116–123, 1954.
26. Haynes, M. K.: Multidimensional Magnetic Memory Selection Systems, *IRE Trans. on Electronic Computers*, **EC-1**:25–32 (December, 1952).
27. Herr, R.: Mixed Ferrites for Recording Heads, *Electronics*, **24**:124–125 (April, 1951).
28. Hoagland, A. S.: A Logical Reading System for Non-return-to-zero Magnetic Recording, *IRE Trans. on Electronic Computers*, **EC-4** (3): 93–95 (September, 1955).

29. Hoagland, A. S.: High Resolution Magnetic Recording Structures, *IBM J. Research Development*, **2** (2): 90–104 (April, 1958).
30. Hoagland, A. S.: Magnetic Drum Recording of Digital Data, *Trans. AIEE*, **73**:381–385 (September, 1954).
31. Hoagland, A. S.: Magnetic Recording Head Design, *Proc. Western Joint Computer Conf.*, San Francisco: 26–31, Feb. 7–9, 1956.
32. Hoagland, A. S.: Magnetic Data Recording Theory: Head Design, *Trans. AIEE*, part I, **75**:506–512 (November, 1956).
33. Holden, A. N., B. T. Matthias, W. J. Merz, and J. P. Remeika: New Class of Ferroelectrics, *Phys. Rev.*, **98**:546 (Apr. 15, 1955).
34. Hollander, Gerhard L.: Design Fundamentals of Photographic Data Storage, *Proc. WESCON Computer Sessions*, Los Angeles: 44–49, Aug. 25–27, 1954.
35. Hollander, Gerhard L.: Quasi-random Access Memory Systems, *Proc. Eastern Joint Computer Conf.*, New York, T-92: 128–135, Dec. 10–12, 1956.
36. Holt, A. W.: A Preliminary Report on the Prototype Diode-capacitor Memory, *Proc. Argonne Conference on Digital Computers*, August, 1953.
37. Holt, A. W.: An Experimental Rapid Access Memory Using Diodes and Capacitors, *Proc. Assoc. Computing Machinery*, Toronto Meeting: 133–141, September, 1952.
38. Horn, I.: An Emitter-follower-coupled, High-speed Binary Counter, 1958 *IRE WESCON Convention Record*, part 4, Los Angeles: 54–61, Aug. 19–22, 1958.
39. Jaynes, E. T.: *Ferroelectricity*, Princeton University Press, Princeton, N.J., 1953.
40. Karan, C., B. J. Skinner: $BaTiO_3$-KF Phase Diagram, *J. Chem. Phys.*, **21**:2225 (1953).
41. Klein, Rudolph J.: Automatic Beam Current Stabilization for Williams Tube Memories, *IRE Trans. on Electronic Computers*, **EC-2** (4): 8–11 (December, 1953).
42. Kodis, R. D., S. Ruhman, and W. D. Woo: Magnetic Shift Register Using One Core per Bit, *IRE Convention Record*, part 7, New York: 38–42, 1953.
43. Lawrance, R. B., R. E. Wilkins, and R. A. Pendleton: Apparatus for Magnetic Storage on Three-inch Wide Tapes, *Proc. Eastern Joint Computer Conf.*, New York, T-92: 84–89, Dec. 10–12, 1956.
44. Lawrence, W. W., Jr.: Recent Developments in Very-high-speed Magnetic Storage Techniques, *Proc. Eastern Joint Computer Conf.*, New York, T-92: 101–103, Dec. 10–12, 1956.
45. Lubkin, Samuel: An Improved Reading System for Magnetically Recorded Digital Data, *IRE Trans. on Electronic Computers*, **EC-3** (3): 22–25 (September, 1954).
46. Mason, W. P., and B. T. Matthias: Theoretical Model for Explaining the Ferroelectric Effect in Barium Titanate, *Phys. Rev.*, **74**:1622–1636 (Dec. 1, 1948).
47. Matthias, B. T., C. E. Miller, and J. P. Remeika: Ferroelectricity of Glycine Sulfate, *Phys. Rev.*, **104**:849–850 (Nov. 1, 1956).
48. Maughmer, J. M., and H. D. Huskey: A Study of Refill Phenomena in Williams Tube Memories, *IRE Trans. on Electronic Computers*, **EC-7** (1): 23–31 (March, 1958).
49. McGuigan, J. H.: Combined Reading and Writing on a Magnetic Drum, *Proc. IRE*, **41**:1438–1444 (October, 1953).
50. Megaw, H. D.: Origin of Ferroelectricity in Barium Titanate and Other Perovskite-type Crystals, *Acta Cryst.*, **5**:739–749 (1952).
51. Merz, W. J.: Domain Formation and Domain Wall Motions in Ferroelectric $BaTiO_3$ Single Crystals, *Phys. Rev.*, **95**:690–698 (August, 1954).
52. Merz, W. J., and J. R. Anderson: Ferroelectric Storage Devices, *Bell Telephone Lab. Record*, September, 1955.
53. Milch, A.: Bit Storage via Electro-optical Feedback, *IRE Trans. on Electronic Computers*, **EC-4** (4): 136–144 (December, 1955).
54. Mutter, W. E.: Improved Cathode-ray Tube for Application in Williams Memory System, *Electrical Eng.*, **71** (4): 352–356 (April, 1952).
55. *Natl. Bur. Standards (U.S.) Tech. News Bull.*, no. 11, November, 1953.
56. *Natl. Bur. Standards (U.S.) Circ.* 551, Computer Development (SEAC and DYSEAC) at the National Bureau of Standards, Washington, D.C., January, 1955.
57. Newhouse, V. L., and N. S. Prywes: High-speed Shift Registers Using One Core per Bit, *IRE Trans. on Electronic Computers*, **EC-5** (3): 114–120 (September, 1956).
58. Papian, W. N.: The MIT Magnetic-core Memory, *Proc. Eastern Joint Computer Conference*, Washington, D.C.: 37–42, Dec. 8–10, 1953.
59. Perry, G. H.: Improvements in Cathode Ray Tube Storage: Application to a Parallel Type of Digital Computer, *Nature*, **168**:372–373 (1951).
60. Pohm, A. V., and S. M. Rubens: A Compact Coincident-current Memory, *Proc. Eastern Joint Computer Conf.*, New York, T-92: 120–123, Dec. 10–12, 1956.
61. Potter, J. T., and P. C. Michel: High Density Digital Recording System, *IRE Trans. on Electronic Computers*, **EC-1**:60–72 (December, 1952).
62. Pulvári, C. F.: Ferroelectrics and Their Memory Applications, *IRE Trans. on Component Parts*, **CP-3** (1): 3–11 (March, 1956).

63. Pulvári, C. F., and G. E. McDuffie, Jr.: Scanners for Ferroelectric Memory Capacitors, *IRE Trans. on Electronic Computers*, **EC-7** (1): 34–40 (March, 1958).
64. Pulvári, C. F.: The Transpolarizer: An Electrostatically Controlled Circuit Impedance with Stored Setting, *Proc. IRE*, **47**:1117–1123 (June, 1959).
65. Pulvári, C. F.: Memory Matrix Using Ferroelectric Condensers as Bistable Elements, *J. Assoc. Computing Machinery*, **2** (3): 169–185 (July, 1955).
66. Raffel, J., and S. Bradspies: Experiments on a Three-core Cell for High-speed Memories, *IRE Convention Record*, part 4, New York: 64–69, 1955.
67. Rajchman, J. A.: Ferrite Apertured Plate for Random-access Memory, *Proc. Eastern Joint Computer Conf.*, New York, T-92: 107–115, Dec. 10–12, 1956.
68. Rajchman, J. A., and A. W. Lo: The Transfluxor—A Magnetic Gate with Stored Variable Setting, *RCA Rev.*, **16**:303–311 (June, 1955).
69. Remeika, J. P.: A Method for Growing Barium Titanate Single Crystals, *J. Am. Chem. Soc.*, **76**:940–941 (1954).
70. Rumble, W. G., and C. S. Warren: Coincident Current Applications of Ferrite Apertured Plates, 1958 *IRE WESCON Convention Record*, part 4, Los Angeles: 62–65, Aug. 19–22, 1958.
71. Ryan, R. D.: A Permanent High Speed Store for Use with Digital Computers, *IRE Trans. on Electronic Computers*, **EC-3** (3): 2–5 (September, 1954).
72. Shirane, G., F. Jona, and R. Pepinsky: Some Aspects of Ferroelectricity, *Proc. IRE* **43**:1738–1793 (December, 1955).
73. Skov, R. A.: Pulse Time Displacement in High-density Magnetic Tape, *IBM J. Research Development*, **2** (2): 130–141 (April, 1958).
74. Slade, A. E., and H. O. McMahon: A Cryotron Catalog Memory System, *Proc. Eastern Joint Computer Conf.*, New York, T-92: 115–119, Dec. 10–12, 1956.
75. Spratt, H. G. M.: Magnetic Recording Tape, *Wireless World*, **57**:88–91 (March, 1951)
76. Temme, D. H.: A Ferroelectric Amplifier and Memory Circuit as Duals of Magnetic Devices, Master's thesis, Department of Electrical Engineering, MIT, May 23, 1955
77. Thorensen, R.: An Improved Cathode Ray Tube Storage System, *Proc. Western Computer Conf.*, Los Angeles: 167–173, Feb. 4–6, 1953.
78. Thorensen, R., and W. R. Arsenault: A New Nondestructive Read for Magnetic Core, *Proc. Western Joint Computer Conf.*, Los Angeles: 111–116, Mar. 1–3, 1955.
79. Tracy, R. A.: Megabit Memory, *Proc. Eastern Joint Computer Conf.*, New York, T-92: 104–106, Dec. 10–12, 1956.
80. Triebwasser, S.: Study of the Second-order Ferroelectric Transition in Tri-glycine Sulfate, *IBM J. Research Development*, **2** (3): 212–217 (July, 1958).
81. Van Sant, O. J., Jr.: Considerations for the Selection of Magnetic Core Materials for Digital Computer Elements, *IRE Convention Record*, part 4, New York: 109–113, 1954
82. von Hippel, A.: Ferroelectricity, Domain Structure and Phase Transitions of Barium Titanate, *Rev. Modern Phys.*, **22**:221–237 (July, 1950).
83. Wallace, R. L., Jr.: Reproduction of Magnetically Recorded Signals, *Bell System Tech. J.*, **30**:1146–1173 (October, 1951).
84. Wanlass, C. L.: Static-dynamic Design of Flip-flop Circuits, *IRE Trans. on Electronic Computers*, **EC-1**:6–18 (December, 1952).
85. Westmijze, W. K.: Studies on Magnetic Recording, *Philips Research Repts.*, Eindhoven, Holland, **8**:148–157 (April, 1953); 161–183 (June, 1953); 245–269 (August, 1953); 344–366 (October, 1953).
86. Widrow, Bernard: A Radio-frequency Nondestructive Readout for Magnetic-core Memories, *IRE Trans. on Electronic Computers*, **EC-3** (4): 12–15 (December, 1954).
87. Wieder, H. H.: Retarded Polarization Phenomena in Barium Titanite Crystals, *J. Appl. Phys.*, **27** (4): 413–416 (April, 1956).
88. Wieder, H. H.: Ferroelectric Hysteresis in Barium Titanate Single Crystals, *J. Appl. Phys.*, **26** (12): 1479–1482 (December, 1955).
89. Wier, Joseph M.: A High-speed Permanent Storage Device, *IRE Trans. on Electronic Computers*, **EC-4** (1): 16–20 (March, 1955).
90. Williams, F. C., T. Kilburn, C. N. W. Litting, D. B. G. Edwards, and G. R. Hoffman: Recent Advances in Cathode Ray Tube Storage, *Proc. Inst. Elec. Engrs.*, **100**, part II: 523–539 (1953).
91. Williams, F. C., and T. Kilburn: A Storage System for Use with Binary Digital Computing Machines, *Proc. Inst. Elec. Engrs.*, **96**, part III: 81–100 (1949).
92. Wong, S. Y.: High Density Williams Storage, *IRE Trans. on Electronic Computers*, **EC-4** (4): 156–158 (December, 1955).
93. Wood, E. A.: Polymorphism in Potassium Niobate, Sodium Niobate, and Other ABO Compounds, *Acta Cryst.*, **4**:353–362 (1951).
94. Younker, E. LeRoy: A Transistor-driven Magnetic-core Memory, *IRE Trans. on Electronic Computers*, **EC-6** (1): 14–20 (March, 1957).

Section 13

SWITCHING CIRCUITS

By

DUDLEY A. BUCK,* *Massachusetts Institute of Technology, Cambridge, Mass.*

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

CONTENTS

13.1. Vacuum-diode Coincidence Circuits..... 13–1
13.2. Cathode-follower Coincidence Circuits..... 13–2
13.3. Dual-control-grid Tubes..... 13–3
13.4. Crystal-diode AND and OR Gates..... 13–4
13.5. A Typical Gate-driver Combination..... 13–5
13.6. The EXCLUSIVE OR Circuit. 13–6
13.7. A Coincidence Gate with Memory..... 13–6
13.8. Resistance Switches..... 13–7
13.8.1. Resistance Switch Using Linear Resistors..... 13–7
13.8.2. Resistance Switch Using Nonlinear Resistors..... 13–8
13.9. Diode Switches..... 13–9
13.10. Magnetic Switches..... 13–10
13.11. Ferroelectric Switches..... 13–12
13.12. Superconductive Switches... 13–13
13.13. Parametric-amplifier Logical Devices..... 13–13

13.1. VACUUM-DIODE COINCIDENCE CIRCUITS

Figure 13.1 illustrates the POSITIVE AND OR NEGATIVE OR circuit and Fig. 13.2 shows the NEGATIVE AND OR POSITIVE OR circuit.

Whenever there is a reverse potential on a diode, Fig. 13.3 shows that it is nearly an open circuit, whereas with a forward potential (anode higher than cathode) the resistance is less than 1 kilohm. Thus if (Fig. 13.1) A, B, and C are at the same potential the current in R is divided among the three tubes. With D greater than any of A, B, or C the diodes are conducting and their resistance is less than 1,000 ohms (1 kilohm). If $R \gg 1$ kilohm then D is the same potential as A, B, and C.

If $A \leq B \leq C$ and R_{AF} is the forward resistance of diode A then

$$D = A + \frac{R_{AF}(V - A)}{R + R_{AF}}$$

Thus if $R \gg R_{AF}$ then $D \sim A$. The back resistance of vacuum diodes is so great that B and C can be ignored.

* Deceased.

For Fig. 13.2 the formula for the potential at D becomes

$$D = C - \frac{R_{CF}(C - V)}{R_{CF} + R}$$

where again $A \leq B \leq C$. If $R \gg R_{CF}$ then $D \sim C$.

Gates such as those in Figs. 13.1 and 13.2 can be combined to make a combination of AND and OR gates.

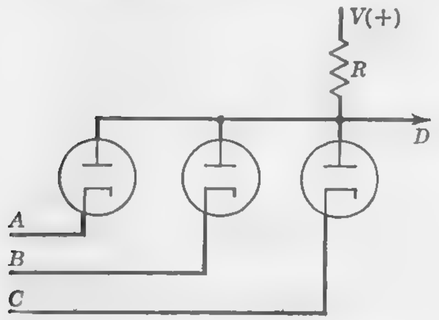


FIG. 13.1. POSITIVE AND circuit.

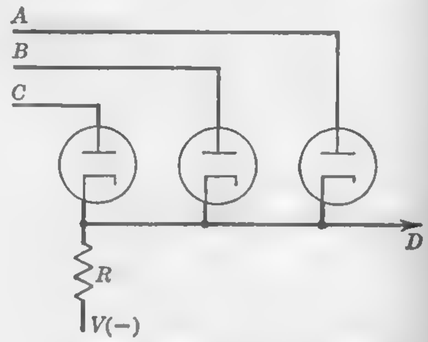


FIG. 13.2. NEGATIVE AND circuit.

The above equations describe the static or d-c performance of such gates. In order to approach the dynamic performance it is necessary to estimate the capacity associated with line D. A reasonable estimate for this is 10 $\mu\mu$f, plus 10 $\mu\mu$f for each tube in the circuit. Lengthy leads or printed circuits will somewhat increase this. When the inputs of Fig. 13.1 swing positive (negative for Fig. 13.2) the diodes tend to cut off and D changes its potential with a time constant RC where C is the associated capacity. With the estimate of C given above, and with R = 100 kilohms, it takes almost 4 μsec for D to change (positive for Fig. 13.1 and negative for Fig. 13.2). In the other direction the rate of change of D depends on the input impedance and the forward resistance of the diodes.

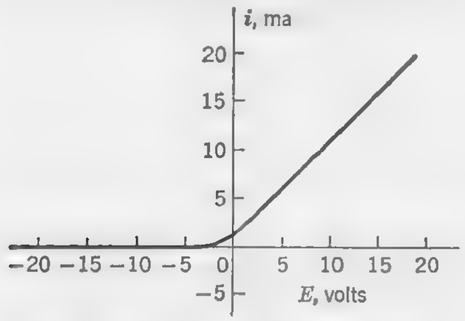


FIG. 13.3. 6AL5 diode characteristic.

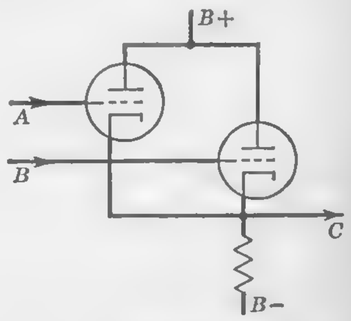


FIG. 13.4. Cathode follower AND circuit.

13.2. CATHODE-FOLLOWER COINCIDENCE CIRCUITS

Cathode-follower circuits (cf. Sec. 11.2) may be used as coincidence circuits. A typical circuit is shown in Fig. 13.4. If either input A or B is relatively high in potential, then C will be high. Coincidence is represented by both A and B being low, producing a low potential at C. This circuit is a NEGATIVE AND and also a POSITIVE OR circuit. Its logical effect is similar to the diode circuit of Fig. 13.2, except that this circuit has power gain.

This circuit has a high-impedance input and a low-impedance output. Performance characteristics can be predicted by using the formulas of Sec. 11.2. Note that

will generally be above A or B in potential by the operating bias of the tube. For example, if a tube passes 10 ma at 5 volts bias and the load resistor R is sufficiently large, then C will be 5 volts above A or B.

This change of level limits the number of cathode followers which may be connected in sequence. Voltage dividers or other circuits must be used to keep voltage levels at the right place.

13.3. DUAL-CONTROL-GRID TUBES

Figures 13.5 and 13.6 show the control-grid and suppressor-grid characteristics of the 6AS6 tube. The 7AK7 was developed specifically for computer applications.

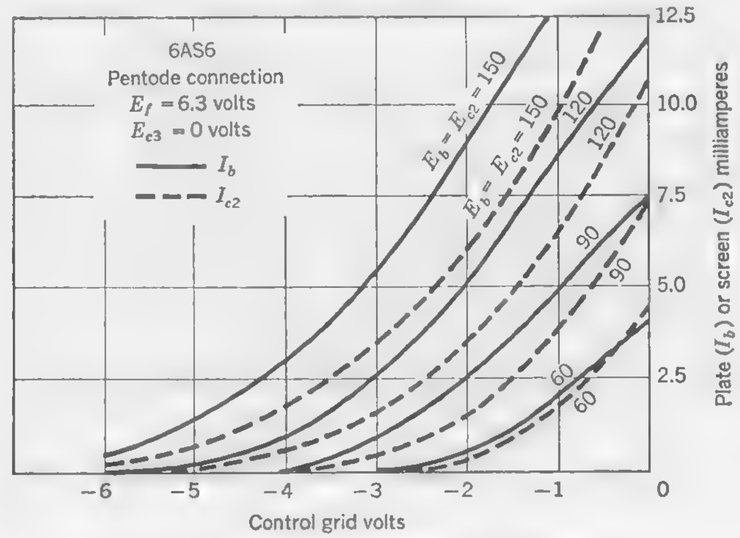


FIG. 13.5. 6AS6 control-grid characteristics.

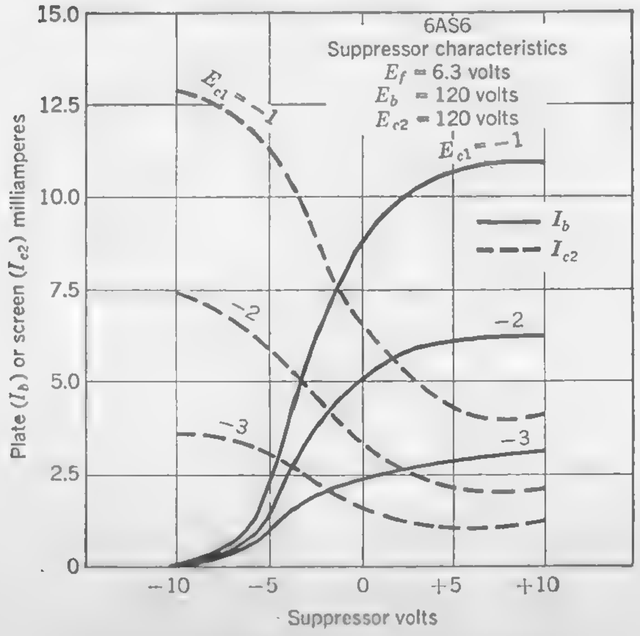


FIG. 13.6. 6AS6 suppressor-grid characteristics.

It has essentially the same characteristics as the 6AS6 except for passing three to four times as much current. A typical 7AK7 circuit is shown in Fig. 13.7.

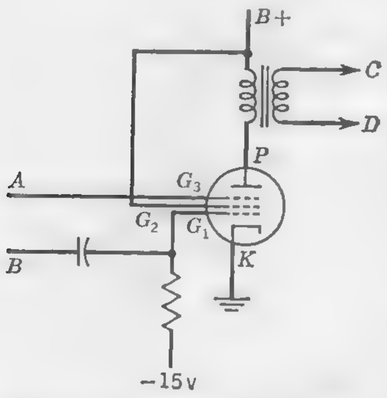


FIG. 13.7. Dual-control-grid gate-drive circuit.

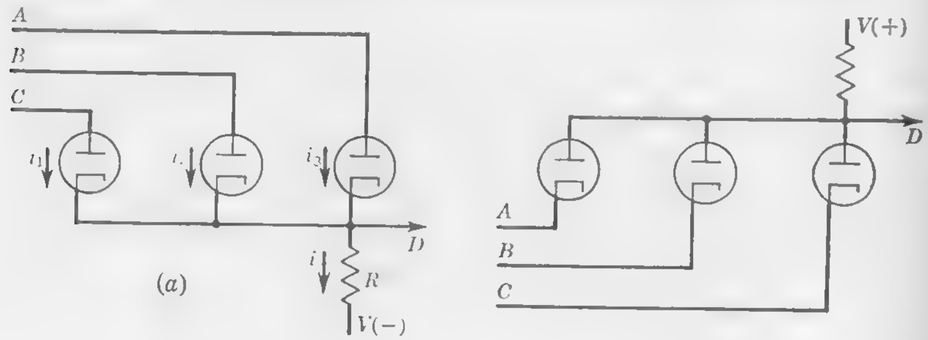
Since the screen dissipation is not so great as the plate, usually the low-duty-cycle signals are connected to grid G_1 (B). If G_1 is high and G_3 low, all the cathode current goes to the screen G_2. The use of low-duty-cycle signals on G_1 is emphasized by the coupling capacitor in line B of Fig. 13.7.

The transformer provides a positive output at C if D is connected to ground.

13.4. CRYSTAL-DIODE "AND" AND "OR" GATES

Figure 13.8 illustrates crystal-diode gates analogous to Figs. 13.1 and 13.2. Part a of Fig. 13.8 illustrates a NEGATIVE AND or a POSITIVE OR circuit, whereas part b illustrates a POSITIVE AND or a NEGATIVE OR circuit.

In a the fall time of D, and in b the rise time, depends on resistor R and the distributed capacity. Thus shorter time constants require smaller values of R. But



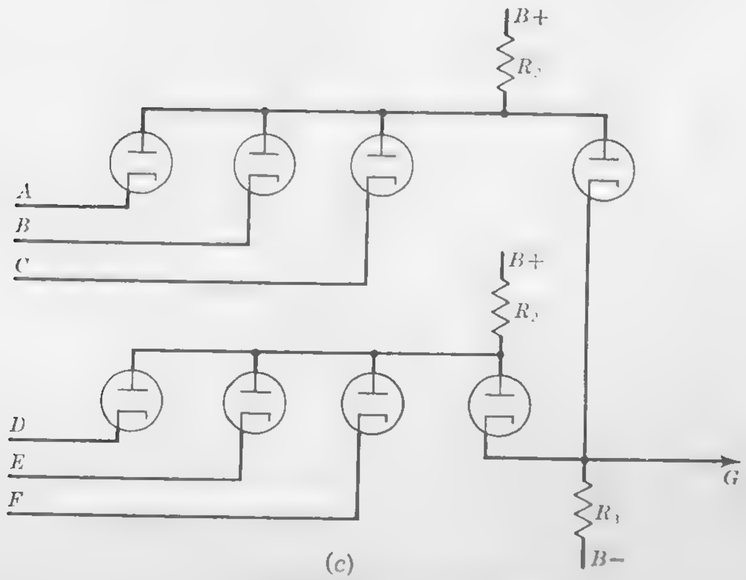


FIG. 13.8. Crystal-diode gates.

small values of R imply greater steady-state currents or more power dissipation and require lower-impedance input drivers.

If it is assumed that no current flows to D, then the currents in Fig. 13.8a must satisfy.

$$i = i_1 + i_2 + i_3 \qquad i_2R_{x2} = B\text{-}D$$
$$iR = D\text{-}V \qquad i_3R_{x3} = A\text{-}D$$
$$i_1R_{x1} = C\text{-}D$$

where R_{xi} is the resistance of the respective crystal.

If one of A, B, or C, say A, is substantially greater in potential than the other inputs, then R_{x2} and R_{x3} are each roughly 10^4 times R_{x1}. Thus the currents i_2 and i_3 are practically negligible and $i_1 = i$. Usually, the forward resistance is about 100 to 200 ohms and R is much larger, causing D to be only slightly less than A.

Reliable performance of such a gate requires that it still function properly when the back resistances of the crystals are decreased by a substantial factor. For example, suppose that the back resistance is at least 10^6 ohms when new, and reliable operation is required when this has decreased to 100 kilohms. Assume the inputs swing from ground (0 volt) to −20 volt. With two inputs low and one high and new crystals, D will be approximately $-100i$ volt (assuming the forward resistance of the crystal is 100 ohms), and A must supply $i + 0.04$ ma of current. If the back resistance of the crystals is 100 kilohms, A must supply $i + 0.4$ ma of current and D will be approximately the same. Thus reliability of the circuit under these circumstances depends on the source impedance of A.

For more complicated networks it is necessary to write the corresponding equations for current and potentials, decide on the limits in back-resistance deterioration, and then work out the potentials for the worst combination of inputs, taking into account the driving impedance of the inputs.

Figure 13.8c shows two three-input AND gates driving an OR gate. In terms of negative signals this may be interpreted as two three-input OR gates driving an AND gate. In cascading of such circuits deterioration of back resistance becomes even more critical.

13.5. A TYPICAL GATE-DRIVER COMBINATION

Figure 13.9 shows a typical gate-driver combination. The direct input I_4 provides for the possibility of connecting other diodes into the circuit to make a more complicated gate. However, this use would reduce the margins of safety, and a more practical use for the terminal I_4 is as a means of checking the characteristics of the diodes.

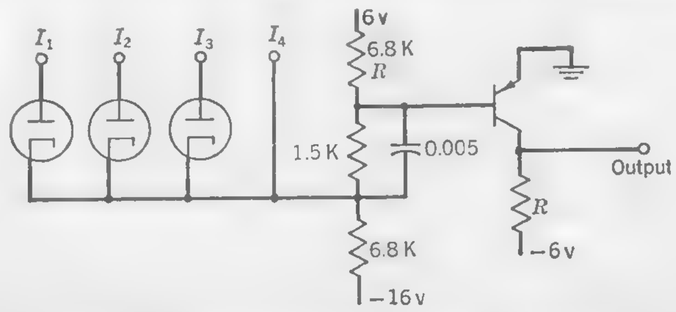


FIG. 13.9. Typical diode gate and driver combination.

The inputs I_1 to I_3 are expected to swing from −0.1 to −6 volt. If all inputs are negative the base of the transistor is low in potential, and current flows at the collector so that the output is high in potential. Thus this circuit inverts the signal very much like its vacuum-tube counterpart. The output is capable of driving five inputs such as I_1 to I_3. The load resistor R may range from 0 to perhaps 3 kilohms, since the diode gates connected to the output terminal supply the actual load.

Two such diode gate-driver circuits may be interconnected to form a flip-flop, as shown in Fig. 13.10. The outputs of the flip-flop I_1 may be connected to other gates, and positive signals at I_2 or I_3 of the respective sides may serve as set and reset signals.

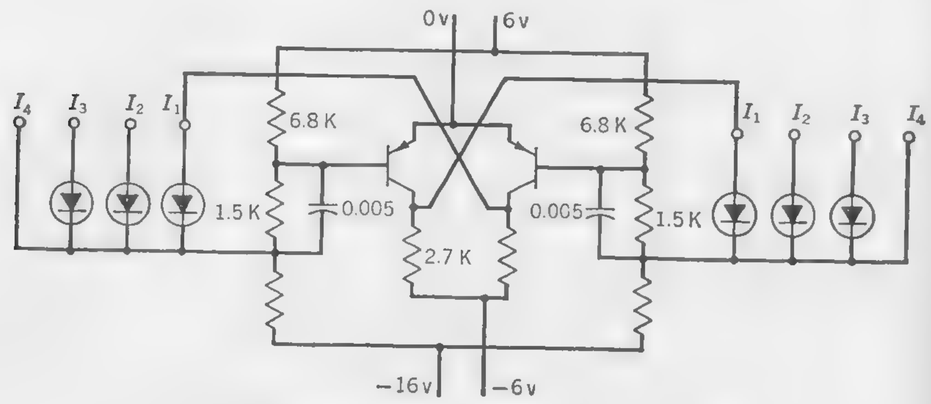


FIG. 13.10. Two gate-drivers connected to form a flip-flop.

13.6. THE EXCLUSIVE OR CIRCUIT

Figure 13.11 shows a transistor difference amplifier followed by a driver. Whenever the two inputs I_1 and I_2 are nominally equal (differ by less than 0.8 volt, say) both transistors conduct and the current is equally divided in the two load resistors. Whenever the inputs are different (differ by at least 1.2 volts, say) one transistor cuts

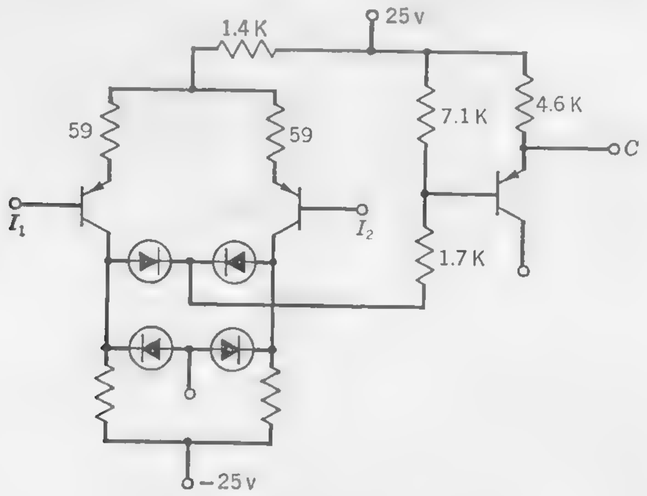


FIG. 13.11. EXCLUSIVE OR circuit.

off and the low signal is transmitted through the OR circuit to the output driver. Small differences of signal or variations of component parameters lead to small changes of current in the voltage divider feeding the output driver. These small fluctuations are absorbed by the clamp diodes.

13.7. A COINCIDENCE GATE WITH MEMORY

Figure 13.12 shows a coincidence gate with memory which was developed at the University of Illinois. If the last inputs are different the last state is retained. If the inputs are the same then the output C follows the input and the output $\neg C$ (not C)

is the complement of the input. The circuit consists of a difference amplifier feeding a Schmidt trigger through two Zener diodes. The voltage drop of the Zener diodes produces two different thresholds at which signals are identified as being the same or different. The Zener diodes may be said to provide a hysteresis effect. The Schmidt trigger is followed by two drivers, one for C and the other for $\neg C$.

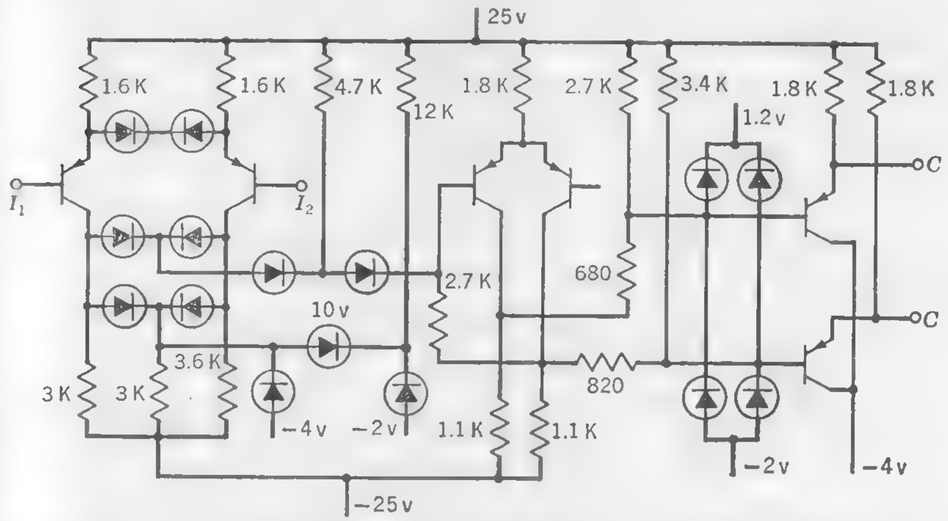


FIG. 13.12. A coincidence gate with memory. (*University of Illinois.*)

Circuits such as that shown in Fig. 13.12 may be designed which will operate in less than 30 mμsec.

13.8. RESISTANCE SWITCHES*

Multiposition switches can be constructed in small sizes using linear resistors and in somewhat larger sizes using nonlinear resistors. The size in both cases is limited by the ability of the device at each output terminal of the switch to discriminate between the voltage representing selection of that terminal and the largest voltage, which should not represent selection. These two voltages become nearly equal as the switch becomes large. The devices at the output terminals are usually biased diodes or biased vacuum tubes.

Figure 13.13 illustrates the basic circuit of a resistance switch having eight output terminals. Three pairs of control wires are required to select among the eight output terminals. One wire of each pair is connected to a voltage V, and the other is grounded. Toggle switches determine which of each pair is at V and which is grounded. In practice, the toggle switches are vacuum tubes or transistor flip-flops, and the two possible switch positions represent the two stable states of each flip-flop.

13.8.1. Resistance Switch Using Linear Resistors. With the switches as shown in Fig. 13.13, terminal 5 is at voltage V, terminals 1, 4, and 7 are at voltage $2V/3$, terminals 0, 3, and 6 are at voltage $V/3$, and terminal 2 is at ground potential. In general, if n is the number of control pairs and 2^n the number of output terminals, there are $n + 1$ possible output voltages: V, $[(n-1)/n]V$, $[(n-2)/n]V$, $[(n-3)/n]V$, . . , 0. The device at each output terminal must discriminate between V and all other voltages, the largest of which is $[(n-1)/n]V$. The voltage difference on which the terminal devices must operate is $V - [(n-1)/n]V = V/n$. For the example of an eight-position switch given above, this difference equals $V/3$ volts. As n becomes large, the voltage difference becomes a small fraction of V, and severe stability requirements are placed on the selecting voltage, the threshold of the terminal devices, and the value of the resistors.

* By Dudley A. Buck.

13.8.2. Resistance Switch Using Nonlinear Resistors. The voltage difference on which the terminal devices must operate can be increased by replacing the linear resistors of Fig. 13.13 with nonlinear resistors such as thyrite. A resistor of this type has a resistance which drops as the current through it increases. The volt-ampere characteristics are given by $I = KV^x$ where K is a constant. The commercially available range of the exponent x is between 2.49 and 3.36.

If the linear resistors of the eight-position switch described above are replaced by nonlinear resistors having $x = 2.8$, the voltage difference on which the terminal devices

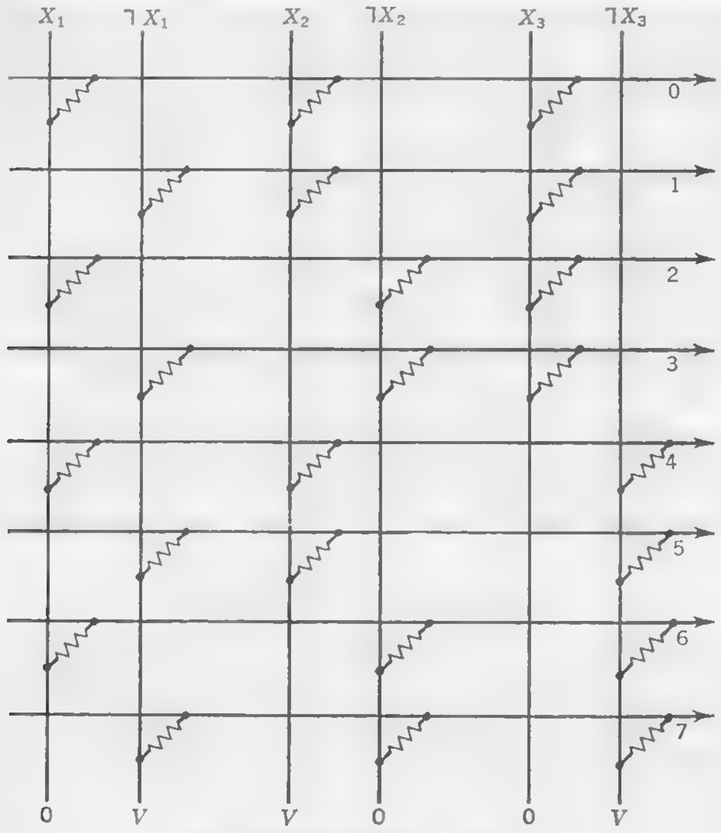


FIG. 13.13. Resistance switch.

must operate is increased from $V/3$ to $V/2.28$. In general, if n is the number of control pairs and 2^n is the number of output terminals, there are $n + 1$ possible output voltages: V, $\frac{(n-1)^{1/x}}{(n-1)^{1/x}+1}V$, $\frac{(n-2)^{1/x}}{(n-2)^{1/x}+2^{1/x}}V$, $\frac{(n-3)^{1/x}}{(n-3)^{1/x}+3^{1/x}}V$, . . . , 0

The difference between V and the next largest of these is $\frac{V}{(n-1)^{1/x}+1}$. This difference is given graphically in Fig. 13.14 as a function of x for several values of n. The portion of each curve for which $x = 1$ (left-hand edge of graph) represents linear-resistance switches. As can be seen, nonlinear resistors effect a dramatic improvement in operating-voltage swing for large switches.

In the above discussion, the internal resistance of the voltage sources which drive the control pairs has been assumed to be zero and the impedance of the load attached to each terminal has been assumed to be infinite. In practice this is not the case. The values of the resistances of the switch should be chosen to be as small as possible consistent with the current capabilities of the control-pair drivers. Lowering the resistance values in a given switch design while keeping the driving voltage constant increases the driving power required by the switch, shortens the time taken to switch, and allows the operating-voltage swing to approach the theoretical values given in

the above discussion. The time taken to switch is that time required to charge the capacitance of the load and the switch plus the stray capacitances involved. Lowering the switch resistance thus lowers the charging resistance, thereby shortening the RC time constant.

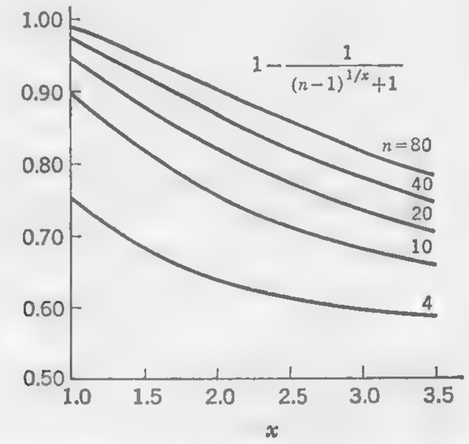


FIG. 13.14. Ratio of highest nonselecting voltage to selecting voltage in nonlinear resistance switches.

13.9. DIODE SWITCHES

The AND and OR gates described in Sec. 13.4 can be used to build switches. A binary eight-position switch is shown in Fig. 13.15. For each of the eight possible sets of values of x_1, x_2, x_3 one of the eight lines 0, 1, 2, . . . , 7 is high.

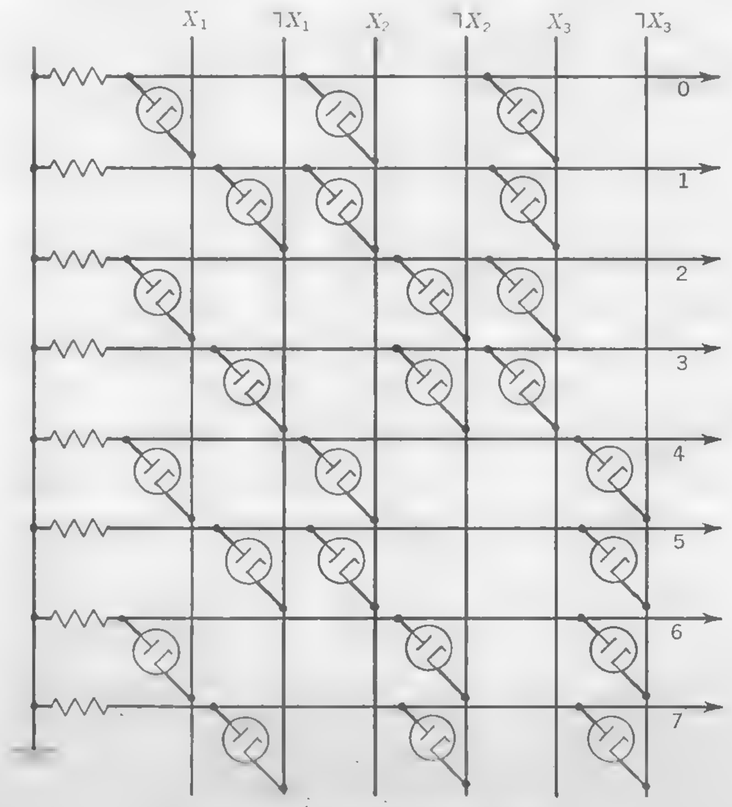


FIG. 13.15. Crystal-diode switch.

13.10. MAGNETIC SWITCHES

Magnetic switches may be made with linear transformers using nonsaturating techniques. Such a switch is shown in Fig. 13.16. The switching function is actually accomplished by the diodes, and the transformers provide impedance matching and/or

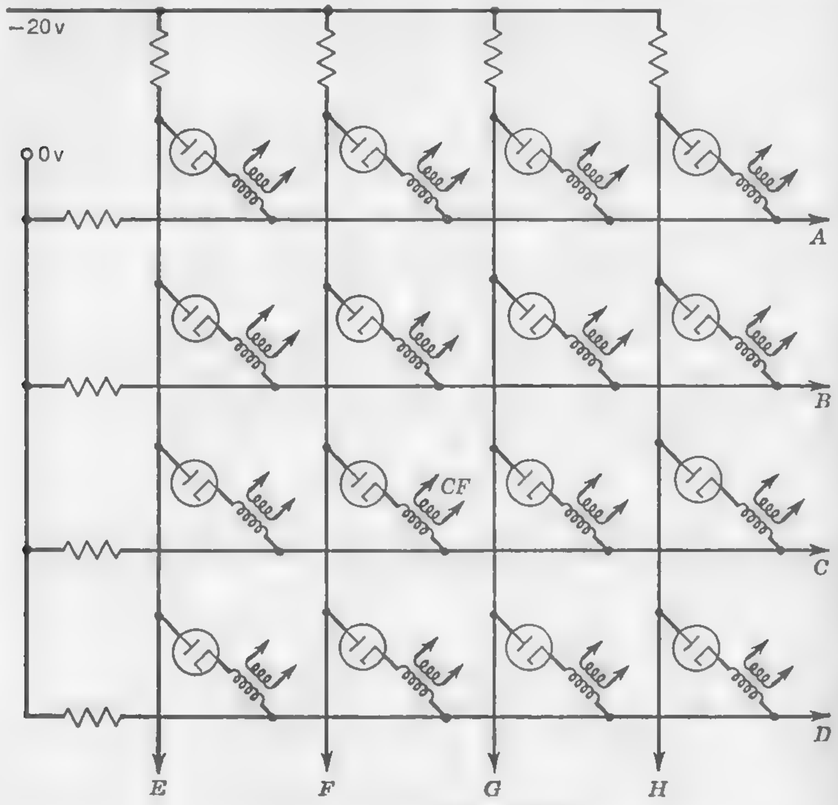


FIG. 13.16. Transformer switch.

isolation. Also total power dissipation may be less than that required in a standard diode switch in that the duty cycle may be relatively low.

With no inputs all the diodes have 20 volts in the backward direction on them. If a line such as F receives a positive pulse of 20 volts (this must be precisely controlled) all the diodes attached to F will have 0 volt across them. If, simultaneously, a negative pulse is applied to a horizontal line such as C, the primary of the transformer CF sees a full 20 volts and the other transformers on the horizontal line C will have 0 volt across their primaries.

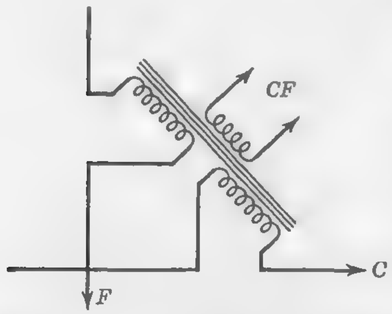


FIG. 13.17. Coincident-current switch transformer.

If a core material with a square hysteresis loop were used (see Fig. 13.18) the diodes could be left out in Fig. 13.16 and two primaries on each transformer could be connected as shown in Fig. 13.17. Then a current i in line F would be insufficient to switch the transformers associated with it. However, if simultaneously a current of i were passed through line C the corresponding transformer would be switched and there would be a signal at CF. After each such output the transformer would have to be reset by negative current pulses on F and C.

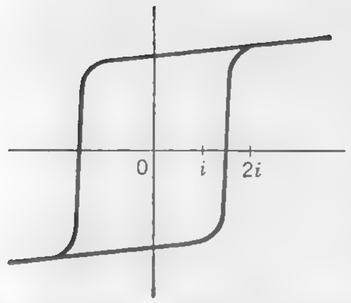


FIG. 13.18. Hysteresis loop for coincident-current transformer.

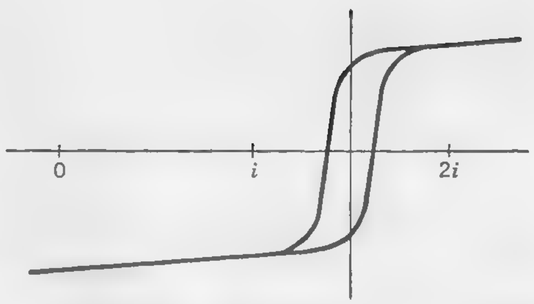
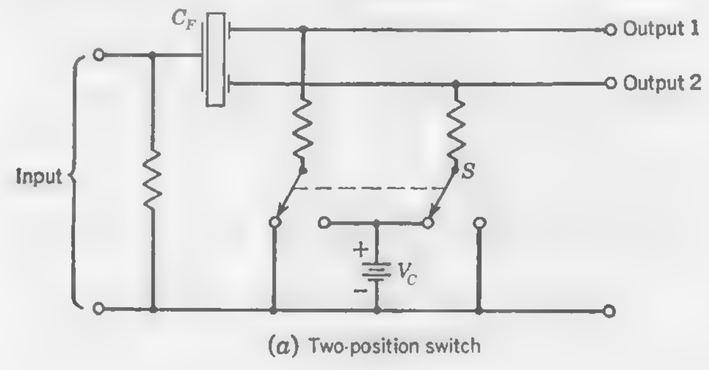
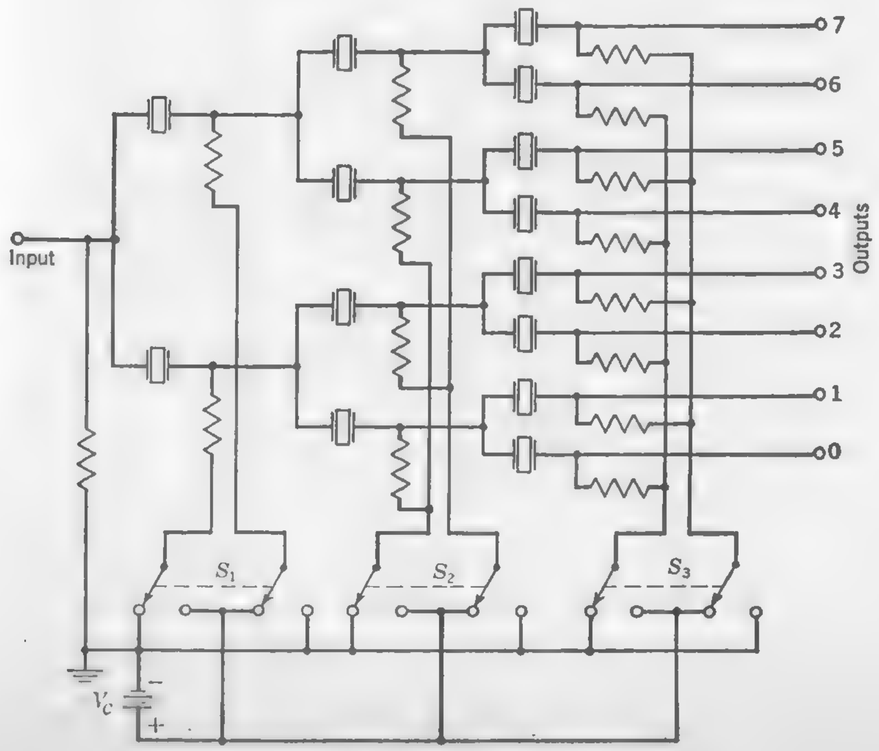


FIG. 13.19. Hysteresis loop for saturating transformer.



(a) Two-position switch



(b) Eight-position switch

FIG. 13.20. Ferroelectric-switch schematic diagram.

A bias winding may be added to the transformer so as to move the operating point to the positions shown in Fig. 13.19. In this mode the flatness of the saturation curves is important and the squareness of the loop is unimportant. Note that in this mode of switching the transformers do not need resetting.

13.11. FERROELECTRIC SWITCHES*

Ferroelectric switches are based on the nonlinear charge-voltage characteristics of capacitors made from materials which exhibit spontaneous alignment of permanent dipole domains. Barium titanate is the most used of these materials at present and

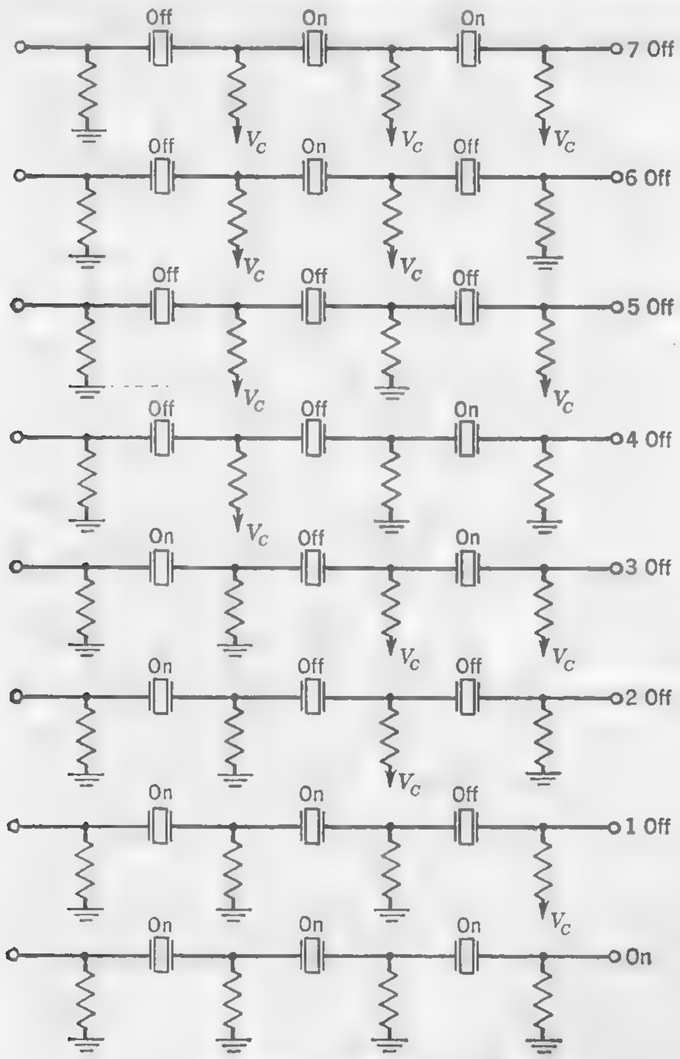


Fig. 13.21. Eight-position-switch analysis.

is available in two forms: sintered ceramics and grown single crystals. The latter have nearly rectangular hysteresis loops.

An eight-position ferroelectric switch is shown in Fig. 13.20. A voltage pulse or an a-c voltage at the input terminal is distributed to one of the eight output terminals as dictated by the voltages on the control wires. Control-wire pairs are driven by binary devices, here represented by toggle switches S_1, S_2, and S_3, in such a way

* By Dudley A. Buck.

that one wire of each pair is grounded and the other is held at a voltage V which exceeds the coercive voltage of the ferroelectric capacitors. With the switches set in a given pattern, all paths except one have at least one capacitor biased in a saturated region where the polarization cannot change. The single path whose three capacitors are unbiased is the selected path. When the input terminal is excited, all three capacitors in this path traverse their hysteresis loops, thus allowing a current to flow between input and selected output.

The eight possible paths are shown in Fig. 13.21 with a given capacitor described as OFF if it is biased into saturation and thus unable to pass current, and ON if not biased.

An eight-position ferroelectric switch has been made from a 1-in. square of barium titanate ceramic. All 14 capacitors are formed by electrodes painted on the two sides of the plate and extended so as to make the interconnections between capacitors. Silver-paste electrodes, fired on, are suitable for ceramic materials, but evaporated electrodes are recommended for single crystals.

13.12. SUPERCONDUCTIVE SWITCHES

There are more than 20 metals whose resistance completely disappears at temperatures somewhat above absolute zero. Sufficient magnetic field will cause the resistance to reappear. Such superconductive metals can be used to build devices with current gain, and they provide another possible digital computer component.

Using liquid helium as a suitable environment, a switch can be built by winding a niobium coil around a tantalum core. Such a device has been called a "cryotron."[5] Under such circumstances both the niobium and the tantalum

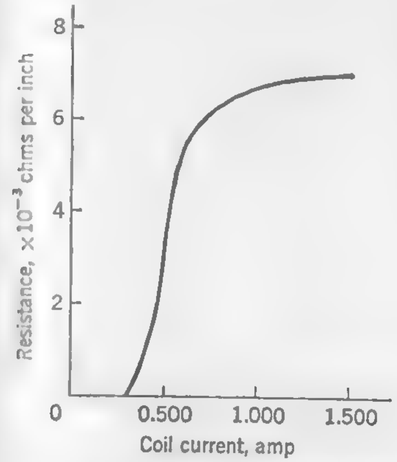


Fig. 13.22. Resistance characteristic of a tantalum cryotron.

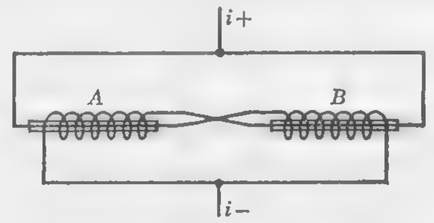


Fig. 13.23. A cryotron flip-flop.

are superconductive. When current is passed through the niobium coil, the tantalum can be switched to "normal" conductivity. The resistance characteristic for the tantalum is shown in Fig. 13.22.

Two such cryotrons may be connected as shown in Fig. 13.23 to form a flip-flop.

The two stable states are with current flowing in the coil of A and the core of B, or in the core of A and the coil of B. Current in the coil of A causes the core to be resistive, inhibiting current in the coil of B. Thus, the core of B is superconductive, encouraging the current in the coil of A. The flow of current must be limited so that the coils always remain superconductive. Other tantalum cores in series with the cores of A and B can be used to set and reset the flip-flops. Likewise, coils in series with the coils of A and B can control other "read-out" cores.

By using thin film techniques submicrosecond speeds may be obtained.

13.13. PARAMETRIC-AMPLIFIER LOGICAL DEVICES

A parametric subharmonic phase-locked oscillator can be used as a computer component. When driven by a pump frequency $2f$ the device will oscillate at a fre-

quency f in one of two phases which differ by 180 deg. Let one phase be called the 0 phase and the other the 1 phase representing, say, a binary 0 and a binary 1.

Information can be transferred through a group of such amplifiers by having suitable coupling interconnections and driving them in three groups using three phase bursts of pump frequencies. The three phases of pump frequencies must occur as shown in Fig. 13.24. Each amplifier will begin to oscillate as the pump frequency is

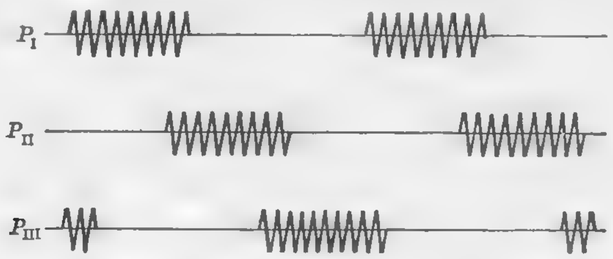


FIG. 13.24. Three phases of pump frequencies. (*R.C.A.*)

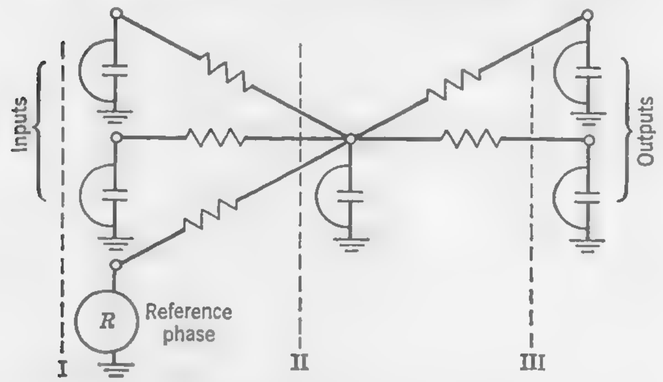


FIG. 13.25. Reference input. (*R.C.A.*)

applied, and the phase (0 or 1) will depend upon the input signals. A number of inputs may be combined in a resistance adder, and the phase of the oscillation will coincide with the phase which appears on the majority of the inputs.

Two input AND and OR gates may be formed by using a third reference input as shown in Fig. 13.25. The phase established at C for various conditions is shown in the following table.

Phase at		Reference	Induced phase at C
A	B		
Zero	Zero	Zero	Zero
One	Zero	Zero	Zero
Zero	One	Zero	Zero
One	One	Zero	One
Zero	Zero	One	Zero
One	Zero	One	One
Zero	One	One	One
One	One	One	One

Thus, with a reference phase of 0 it is an AND circuit, but with a reference phase of 1 it is an OR circuit.

At microwave frequencies two-phase pump systems can be used by using so-called isolators which have high ratios of forward to backward transmission.

Using a continuous pump frequency it is possible to switch the phase of the parametric amplifier forcibly from 0 to 1 or 1 to 0. The fact that this phenomenon is amplitude-sensitive to the input signal has suggested the possibilities of coincidence switching and the use of parametric amplifiers in memory arrays.

References

1. Andrews, L. J.: A Technique for Using Memory Cores as Logical Elements, *Proc. Eastern Joint Computer Conf.*, New York, **T-92**:39–46, Dec. 10–12, 1956.
2. Angell, James B.: High-temperature Silicon-transistor Computer Circuits, *Proc. Eastern Joint Computer Conf.*, New York, **T-92**:54–57, Dec. 10–12, 1956.
3. Beter, Ralph H., William E. Bradley, and Ralph B. Brown: Surface-barrier Transistor Switching Circuits, *IRE Convention Record*, part 4, New York: 139–145, 1955.
4. Booth, G. W., and T. P. Bothwell: Logic Circuits for a Transistor Digital Computer, *IRE Trans. on Electronic Computers*, **EC-5**(3):132–138 (September, 1956).
5. Buck, D. A.: A Magnetically Controlled Gating Element, *Proc. Eastern Joint Computer Conf.*, New York, **T-92**:47–50, Dec. 10–12, 1956.
6. Constantine, G., Jr.: A Load-sharing Matrix Switch, *IBM J. Research Development*, **2**(3):204–211 (July, 1958).
7. Dickinson, W. E., and R. M. Walker: Reliability Improvement by the Use of Multiple-element Switching Circuits, *IBM J. Research Development*, **2**(2):142–147 (April, 1958).
8. Domenico, R. J.: Simulation of Transistor Switching Circuits on the IBM 704, *IRE Trans. on Electronic Computers*, **EC-6**(4):242–246 (December, 1957).
9. Duijvestijn, A. J. W.: On the Transition from Superconducting to Normal Phase, Accounting for Latent Heat and Eddy Currents, *IBM J. Research Development*, **3**(2):132–139 (April, 1959).
10. Dunham, B., D. Middleton, J. H. North, J. A. Sliter, and J. W. Weltzien: The Multipurpose Bias Device—part II: The Efficiency of Logical Elements, *IBM J. Research Development*, **3**(1):46–53 (January, 1959).
11. Easley, James W.: Transistor Characteristics for Direct-coupled Transistor Logic Circuits, *IRE Trans. on Electronic Computers*, **EC-7**(1):6–16 (March, 1958).
12. Harris, J. R.: Direct-coupled Transistor Logic Circuitry, *IRE Trans. on Electronic Computers*, **EC-7**(1):2–6 (March, 1958).
13. Henle, R. A.: High-speed Transistor Computer Circuit Design, *Proc. Eastern Joint Computer Conf.*, New York, **T-92**:64–66, Dec. 10–12, 1956.
14. Hogue, E. W.: A Saturable-transformer Digital Amplifier with Diode Switching, *Proc. Eastern Joint Computer Conf.*, New York, **T-92**:58–64, Dec. 10–12, 1956.
15. Holden, A. N., B. T. Matthias, W. J. Merz, and J. P. Remeika: New Class of Ferroelectrics, *Phys. Rev.*, **98**:546 (Apr. 15, 1955).
16. Hooton, J. A., and W. J. Merz: Etch Patterns and Ferroelectric Domains in $BaTiO_3$ Single Crystals, *Phys. Rev.*, **98**:409–413 (Apr. 15, 1955).
17. Huang, C., E. Slobodzinski, and B. White: Transistor Shift Registers, *IRE Convention Record*, part 4, New York: 140–144, 1954.
18. Kudlich, R. A.: A Set of Transistor Circuits for Asynchronous, Direct-coupled Computers, *Proc. Western Joint Computer Conf.*, Los Angeles: 124–129, Mar. 1–3, 1955.
19. Little, E. A.: Dynamic Behavior of Domain Walls in $BaTiO_3$, *Phys. Rev.*, **98**:978 (May, 1955).
20. Marcus, M. P.: Doubling the Efficiency of the Load-sharing Matrix Switch, *IBM J. Research Development*, **3**(2):194–197 (April, 1959).
21. Mason, W. P., and R. F. Wick: Ferroelectrics and the Dielectric Amplifier, *Proc. IRE*, **42**:1606–1620 (November, 1954).
22. Merz, W. J.: Double Hysteresis Loop of $BaTiO_3$ at the Curie Point, *Phys. Rev.*, **91**:513–517 (Aug. 1, 1953); **95**:690–698 (Aug. 1, 1954).
23. Mitsui, T., and J. Furuichi: Domain Structure of Rochelle Salt and KH_2PO_4, *Phys. Rev.*, **90**:193–202 (Apr. 15, 1953).
24. Prom, G. J., and R. L. Crosby: Transistor Switching Circuits for High-speed Computer, *IRE Trans. on Electronic Computers*, **EC-5**(4):192–196 (December, 1956).
25. Rajchman, J. A., and H. D. Crane: Current Steering in Magnetic Circuits, *IRE Trans. on Electronic Computers*, **EC-6**(1):21–30 (March, 1957).
26. Reeber, M. D.: Geometric Effects in the Superconducting Transition of Thin Films, *IBM J. Research Development*, **3**(2):140–146 (April, 1959).
27. Schwertz, F. A., and R. T. Steinbach: Nonlinear Resistors in Logical Switching Circuits, *Proc. Western Computer Conf.*, Los Angeles: 174–186, Feb. 4–6, 1953.

28. Seader, L. D.: Magnetic-recording-head Selection Switch, *IBM J. Research Development*, **2**(1):36–42 (January, 1958).
29. Sebestyen, George: A Design Technique for Pedestal-free Switching Circuits, *IRE Trans. on Electronic Computers*, **EC-6**(3):162–166 (September, 1957).
30. Shevel, W. L., Jr.: Observations of Rotational Switching in Ferrites, *IBM J. Research Development*, **3**(1):93–95 (January, 1959).
31. Smith, W. V., J. Overmeyer, and B. A. Calhoun: Microwave Resonance in Gadolinium iron Granet Crystals, *IBM J. Research Development*, **3**(2):153–162 (April, 1959).
32. Sterzer, F., and D. Blattner: Fast Microwave Logic Circuits, *IRE National Convention Record*, part 4, New York: 252–258, March, 1959.
33. von Hippel, A. R.: *MIT Lab. for Insulation Research Tech. Rept.* 99, p. 38, August 1955.
34. Wanlass, C. L.: Transistor Circuitry for Digital Computers, *IRE Trans. on Electronic Computers*, **EC-4**(1):11–16 (March, 1955).
35. Young, Frederick H.: Analysis of Shift Register Counters, *J. Assoc. Computing Machinery*, **5**(4):385–388 (October, 1958).
36. Zimbel, Norman: Packaged Logical Circuitry for a 4-Mc Computer, *IRE Convention Record*, part 4, New York: 133–139, 1954.

Section 14

INFORMATION CODING AND SWITCHING THEORY

By

RICHARD W. HAMMING, *Member Technical Staff, Bell Telephone Laboratories, Inc., Murray Hill, N.J.*

DAVID SLEPIAN, *Member Technical Staff, Bell Telephone Laboratories, Inc., Murray Hill, N.J.*

ARTHUR W. BURKS, *Professor, Department of Philosophy, University of Michigan, Ann Arbor, Mich.*

CONTENTS

14.1. Binary Coding..... 14–1
14.2. Coded Decimal..... 14–2
14.3. Redundancy..... 14–3
14.4. Error-detecting Codes..... 14–4
14.5. Use of Redundant Codes.... 14–4
14.6. On Choosing a Redundant Code..... 14–5
14.7. The Geometrical Model..... 14–5
14.8. Detection Methods..... 14–6
14.9. Group Alphabets (Parity-check Alphabets)..... 14–7
14.10. Generalized Hamming Alphabets..... 14–8
14.11. Other Group Alphabets..... 14–10
14.12. Reed-Muller Alphabets..... 14–11
14.13. Bounded-distance Detectors. 14–12
14.14. Plotkin Alphabets..... 14–12
14.15. The Balser-Silverman-Wagner Channel and Detector... 14–12
14.16. The Propositional Calculus.. 14–13
14.17. Boolean Algebra..... 14–14
14.18. Truth Tables..... 14–14
14.19. Disjunctive Normal Forms.. 14–15
14.20. Testing for Validity..... 14–15
14.21. Duality..... 14–16
14.22. Using Logic to Design Circuits 14–17
14.23. Minimality for a Single Function..... 14–18
14.24. Complete Decoding Nets.... 14–20
14.25. Minimality for Complete Decoding Nets..... 14–22
14.26. Nets with Delays..... 14–22

Sections 14.1 to 14.15 are on information coding and Secs. 14.16 to 14.26 are concerned with switching theory. See also Sec. 15 on Digital-computer Arithmetic.

14.1. BINARY CODING*

Numbers are stored in many places in a digital computer, for short as well as long periods of time. Many devices are used for such storage, and almost all of them are

* Sections 14.1 to 14.15 are by Richard W. Hamming and David Slepian.

essentially "two-state" devices: on or off, up or down, plus or minus voltage, N-S or S-N magnetization, etc. This fact leads to the use of the binary system of representation of numbers when discussing the mathematical aspects of computers.

The basic idea of the binary system is that of representing numbers in the base-2 system. Almost all the difficulties of the binary system vanish if one pauses to examine the corresponding situation in the decimal system. For example, in the decimal system the integers are represented typically by a number such as

$$1308 \equiv 1 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 8 \times 10^0$$

In a similar way, a typical integer in the binary system is represented in the form

$$101101 \equiv 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

In the decimal system the coefficients of the powers of 10 may range from 0 through 9, while in the binary system the coefficients of the powers of 2 may be only 0 or 1. For numbers less than 1 the decimal system uses such forms as

$$0.176 \cdots \equiv 1 \times 10^{-1} + 7 \times 10^{-2} + 6 \times 10^{-3} + \cdots$$

while the binary system correspondingly uses

$$0.101 \cdots \equiv 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + \cdots$$

Corresponding to decimal arithmetic there is, of course, a binary arithmetic for combining binary numbers. Whole books have been written elucidating the mysteries of the binary arithmetic, but usually a simple comparison with the decimal system will reveal how to handle binary problems (including the conversion from one system of representation to the other).

Logical information is frequently used in a computer. In most cases Boolean algebra is used to handle logical relationships so that the binary representation comes naturally in this case.

Finally, other forms of information, orders or perhaps the names on a payroll, are used in a computer to control the computation on a particular problem. Here, again, the binary form of representation is often used, though usually there is no simple way of combining various pieces of information.

The use of n binary digits allows one to represent 2^n different things. Thus 10 binary digits (called "bits" for short) allows

$$2^{10} = 1024 \sim 10^3$$

items to be represented. This leads to the handy formula that $1\frac{0}{3} = 3\frac{1}{3}$ bits are equivalent to one decimal digit. In order to handle M different things one needs to use at least n binary digits, where n is the least integer, such that

$$n \geq \log_2 M$$

14.2. CODED DECIMAL

From an engineering point of view the binary system is more efficient and easier to design circuits for, but it is unfortunately true that the human race has used the decimal system for a long time and that the schools still teach it. The impasse that results is like that of the adoption of the metric system in place of the English system of measures, the adoption of a "perpetual calendar" in place of our present messy one, or the adoption of an artificial language for universal use in place of our babble of tongues. Logic is all on one side but that does not suffice to win. It seems likely, therefore, that the decimal system will continue to be used for a long time and that for many purposes decimal machines will be built.

The reliable elementary circuits that can be built easily are almost all (in the present state of technology) basically binary. Thus one is forced to represent the decimal digits in some binary form.

One of the simplest forms is the straight representation of the decimal digits 0 through 9 by their equivalent representations in the binary form:

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

In the arithmetic unit of such a computer the "carry" from one decimal column to the next must be arranged to happen whenever the number

$$10 \equiv 1010$$

occurs and at the same time the digits 1010 must be reset to 0000.

In this representation subtraction is the main headache since it is customary when subtracting to convert the number to its "complement form" and then add. The "excess-three" system was devised to get around this trouble by adding three to each binary representation:

0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Thus 0 and 9 are "complements," 1 and 8, etc. An additional advantage is that 0 is now a definite signal and not merely the absence of one. The "carry" in the "excess-three" system is also easier.

There are other systems in use as well as these two. One such preserves the above complement relation by leaving a gap of six binary numbers between the representations of 4 and 5. Systematic studies of the problem of nonredundant representation of the decimal numbers have been made trying to find special features that might be of use in a particular computer. Little of general use has been found beyond those mentioned above.

14.3. REDUNDANCY

Let us suppose we have a binary representation of some kind of information, using n binary digits per unit of information. Let us now adjoin a final binary digit to each set of n digits, choosing the additional digit so that in the $n + 1$ digits there are an even* number of 1's. We now observe that, if for any cause *one* binary digit is changed from a 0 to a 1 or from a 1 to a 0, this change results in an odd number of 1's in the $n + 1$ places. Thus we have a *single*-error-detecting code since any *single* error can be detected.

The extra binary digit is *redundant* since we have used $n + 1$ digits where n would suffice to represent the same information. It should be noted here (and also remem-

* In practice an odd rather than even number of 1's is often preferable. These forms of checks are called "parity checks."

bered later) that one may view the entire $n + 1$ digits as the "message" and not imagine the last one as playing a special role.

14.4. ERROR-DETECTING CODES

The addition of a single check digit is the simplest type of error-detecting code—meaning a form of representation that allows the detection of a single error. This type of code is widely used in computers, especially on magnetic tapes.

Variants of the above are the "2 out of 5" codes. In these codes a single decimal digit is represented by 5 binary digits. Since there are $\binom{5}{2} = 10$ ways of choosing two 1's out of the 5 possible binary digits, one can represent all 10 decimal digits. Different "2 out of 5" codes arise as different correspondences between the decimal digits and their representations are used. These codes are evidently single-error-detecting and will reveal some double errors as well.

Another redundant code that has been used is the one which represents the decimal digit n by the binary number $3n + 2$. This requires only 5 binary digits and is error detecting. Complements are easy to form, and a simple arithmetic unit allows the calculation of the check digit in an independent manner.

An inefficient redundant code that is widely used is the biquinary code for representing decimal digits. The code uses alternately base 2 and base 5. In practice 1 of 2 positions is used for the binary digit and 1 of 5 positions is used for the quinary digit; thus a 2 out of 7 code is used for representing one decimal digit. In particular, the decimal digits are represented by

Decimal	Bi 50	Quinary 43210
0	01	00001
1	01	00010
2	01	00100
3	01	01000
4	01	10000
5	10	00001
6	10	00010
7	10	00100
8	10	01000
9	10	10000

This 2 out of 7 code makes error detection easy. The inefficiency in storing and transmitting information is compensated for in some cases by the ease of designing an error detecting arithmetic unit.

The quibinary code is the obvious variant of the biquinary code.

If enough more redundancy is used it is possible to have a code in which the actual location of an error can be found, and hence in the binary system the error can be corrected. Still more redundancy allows multiple-error detection and correction.

Along these lines is the 8-bit code that represents the decimal digit n by the binary number $27n + 6$. This code allows simple complementing, correction of simple errors, and a relatively simple arithmetic unit that calculates the check bits independently of the other bits.

14.5. USE OF REDUNDANT CODES

Before going into the design and construction of error-detecting and error-correcting codes, it is wise to pause and ask why and where they are used.

The "2 out of 5" codes are the oldest widely used redundant codes, having been used in telephone central offices for a long time. The remarkable results achieved in keeping a large automatic central office going year after year without any serious

breakdowns illustrate the value of error-detecting codes. In practice the codes do not *directly* do much to improve the service since the customer himself makes so many mistakes, but the redundant codes pay off *indirectly* through enabling the maintenance staff to locate and correct malfunctionings before they give much trouble.

Using central-office techniques computers have been built with error detection throughout. When an error occurred, and no one was present, the machine automatically made second and even third trials so that isolated malfunctionings did not halt the computer.

One of the main advantages of an error-detecting code from the coder's point of view is that it not only increases the probability of getting the correct answer to a long problem but also separates the problem-preparation errors from the machine malfunctionings, thus greatly reducing "debugging" troubles.

These remarks on the value of error-detecting codes may be summarized by:

1. Ease of maintenance
2. Reliability of operation
3. Accuracy of results

Error-correcting codes simply go further along these same lines.

14.6. ON CHOOSING A REDUNDANT CODE

The word "noise" occurs frequently in information theory and means roughly all those probabilistically governed perturbations which alter signals in transmission. "Noise" is usually "added" to a signal so that the sum is what is received.

In computer usage "noise" is more definitely associated with failures and distortions of signals. Redundant codes are used to "combat" the noise.

Intelligent design and choice of a redundant code require a knowledge of the kind of noise it is to cope with. It is true that until the computer is running one cannot know the "noise," but careful study of past experiences plus some good imagination can give a fair picture of the noise to be expected.

The realization in actual hardware of a code with its error-detecting and possibly correcting features must also be considered. Thus the biquinary code has been widely used because of the ease of constructing checking and calculating circuits and in spite of the obvious inefficiencies in storage and transmission.

Another example of inefficient coding that is in actual use because of the ease of building the "hardware" is that of forming a rectangular array of the information and then bordering it on two adjacent sides with a row and a column of redundant check bits. These bits are chosen in each case so as to form a simple parity check on its corresponding column or row. Thus the position of a single error is simply given by the row and column that fail to check.

Actually arithmetic itself is somewhat redundant. Thus even with nonredundant codes a decimal arithmetic unit can be partially checked by the familiar "casting out 9's" (or 11's). In the binary system one may use casting out $(2^k \pm 1)$'s to check the arithmetic unit.

The *purpose* the redundant code is to serve must also be considered. In a scientific computing center it is usually better to halt a computation than to guess and go on; in a military computing center it is usually better to guess than to halt.

These two features, the kind of noise to be handled and the type of service to be rendered, must be considered in any discussion of redundant codes and result in there being no simple answer for all situations.

14.7. THE GEOMETRICAL MODEL

In all that follows, we shall suppose that we are given K quantities that are to be represented on the computer by K n-digit binary sequences. The code book by means of which each of the K quantities is associated with one of the K chosen n-digit binary sequences will not concern us here. Rather, attention will be focused on the chosen

binary sequences themselves, and their properties will be studied independently of the quantities they represent. It is convenient to call the collection of chosen n-digit binary sequences a "signaling alphabet of size K"; the individual sequences will be called "letters" of the signaling alphabet.

There is a totality of 2^n n-digit binary sequences. Each sequence can be regarded as specifying a particular vertex of a unit cube in n-space. For example, the sequence 11010 will be said to specify the point in 5-space whose coordinates are (1,1,0,1,0). The 32 possible 5-digit binary sequences specify the 32 vertices of a 5-dimensional cube of unit edge length. A particular signaling alphabet of size K then is specified by designating K vertices of the unit n cube as letters of the signaling alphabet.

Consider now the geometrical interpretation of errors made in transmission of a letter of a signaling alphabet. If a single error is made, the resulting binary sequence differs only in one place from the sent letter. The altered sequence then is represented on the n cube by one of the n vertices that are at a distance of one edge length from the transmitted letter. If two errors are made, the resultant sequence lies two edge lengths away from the sent letter, etc. To discuss these matters further, it is convenient to define the "distance" between two points on the n cube as the minimum number of edges that must be traversed to pass by edges from either point to the other. In terms of sequences, the distance between two n-digit binary sequences is the number of places in which they differ. With this definition we can state that a j-fold error in the transmission of a letter results in a point distant j from the sent letter.

If $K = 2^n$, all vertices of the cube are letters of the signaling alphabet. Any altered letter is again a possible letter and error detection or correction is impossible. If, however, K is sufficiently small compared with 2^n, it may be possible to choose a signaling alphabet so that every two letters are at least distance 2 apart. A single error in transmission of any letter of the alphabet will result in a point of the cube not designated as a letter. Single-error detection is then possible. If all letters are at least distance 3 apart, single-error correction is possible. For, with such an alphabet, a single error in transmission will result in a point closer to the sent letter than to any other letter. More generally, if the minimum distance between letters is $2m + 1$ or more (m a positive integer), all j-fold errors can be corrected, $j = 1, 2, \ldots, m$. Necessary conditions on n, K, and m for such alphabets can be found in Refs. 11 and 13.

14.8. DETECTION METHODS

We have seen that a K-letter signaling alphabet can be regarded as K specially designated vertices of the n cube. We have also observed that because of errors in transmission a sent letter can in general be received as any vertex of the n cube. A detection scheme is a set of rules that associates various vertices of the cube with vertices designated as letters of the alphabet. It is the set of rules by means of which one uses the redundancy of the signaling alphabet to combat the expected errors in transmission.

Very little can be said in the general case regarding how to choose a good signaling alphabet and detection scheme. A system good for one type of noise may be poor for another type of noise. As a simple example, consider the case $K = 2$, $n = 2$. If it is known that transmitted binary digits are disturbed independently with a fixed probability p that a digit be altered, then the signaling alphabet 00, 11 permits detection of a single error. If, however, it is known that successive blocks of two transmitted binary digits are either transmitted correctly or else both digits are altered, then a good signaling alphabet is 00, 10. The detection scheme $00 \rightarrow 00$, $11 \rightarrow 00$, $10 \rightarrow 10$, $01 \rightarrow 10$ which tells how each of the four possible received messages is to be decoded permits error-free transmission.

A situation much studied in information theory, and which may be applicable in computer work, is the following: Errors in different successive blocks of n transmitted binary digits are independent of each other and have a fixed probability distribution. All j-fold errors in a block of length n have probability p_j, $j = 0, 1, \ldots, n$ and $p_0 > p_1 > \cdots > p_n$. If a K-letter n-digit signaling alphabet is being used, all

letters of this alphabet are sent with equal probability $1/K$. In all probability considerations that follow in this chapter, it will be assumed that this situation maintains.

In the case just mentioned, it is possible to describe in a simple manner a detection scheme which minimizes the average probability that a transmitted letter be decoded incorrectly. Such a decoding scheme is called a "maximum-likelihood" detector. The 2^n vertices of the cube are divided into K sets, $R_1, R_2, \ldots, R_K$. Set R_j consists entirely of vertices not closer to any other letter of the alphabet than to the jth letter, $j = 1, 2, \ldots, K$. When a received vertex is a member of R_j, it is decoded as the jth letter of the signaling alphabet, $j = 1, 2, \ldots, K$.

While the best detector in the above sense is known for a given alphabet, for general n and K it is not known how to minimize further the average error probability by proper choice of an alphabet. Since there are only finitely many K-letter n-digit alphabets, a best one can be found in principle by exhaustive methods. For most values of n and K of interest, the number of alphabets is too large for this procedure to be practical. For certain special values of n and K to be discussed later, best alphabets are known.

14.9. GROUP ALPHABETS (PARITY-CHECK ALPHABETS)

Let us define the product of two n-digit binary sequences $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ to be the n-digit binary sequence $a_1 \dotplus b_1, a_2 \dotplus b_2, \ldots, a_n \dotplus b_n$. Here the a's and b's are 0 or 1 and the $\dotplus$ sign means addition modulo 2.

To date, the most extensively studied binary signaling alphabets are those whose letters form an Abelian group under the above-defined law of multiplication. It can be shown that the size of such an alphabet is $K = 2^k$, where k is an integer from the set $0, 1, 2, \ldots, n$. We call such a signaling alphabet an (n,k) alphabet. This class of alphabets includes the Hamming codes and the Reed-Muller codes.

A maximum-likelihood detector for an (n,k) alphabet can be constructed simply by successively forming the rows of Fig. 14.1. Here $I = (0,0, \ldots,0), A_2, A_3, \ldots, A_\mu$

I	A_2	A_3	$\cdots$	A_μ
S_2	S_2A_2	S_2A_3	$\cdots$	S_2A_μ
S_3	S_3A_2	S_3A_3	$\cdots$	S_3A_μ
$\cdot$				
$\cdot$				
$\cdot$				
S_ν	$S_\nu A_2$	$S_\nu A_3$	$\cdots$	$S_\nu A_\mu$

$\nu = 2^{n-k}$, $\mu = 2^k$

FIG. 14.1

are the letters of the (n,k) alphabet. Define the weight of an n-digit binary sequence to be the number of 1's in the sequence. S_j is an n-digit binary sequence of minimum weight not appearing in the first $j - 1$ rows of the table, $j = 2, 3, \ldots, 2^{n-k}$. Column i of the table then lists the sequences of the maximum-likelihood detector that are in the set R_i associated with letter A_i. If there are α_j entries of weight j in the first column of the table, then the probability that any transmitted letter will be decoded correctly is $\Sigma\alpha_j p_j$.

It can be shown that every (n,k) alphabet can be constructed in the following manner: The places (positions) of the letters are divided into two classes: k places (the same ones in each letter) are designated as "information places"; the remaining $n - k$ places are called "check" places. The information places may be occupied by any of the 2^k k-place binary sequences. The entries in the $n - k$ check positions are fixed linear (mod 2) combinations of the entries in the information places. These rules by which the entries in the check places are determined are called "parity checks." Example: in the (5,3) alphabet 00000, 10001, 01101, 00111, 11100, 10110, 01010, 11011, places 1, 2, and 4 (numbered from the left) are the information places. If a gen-

eral letter of the alphabet is $a_1a_2a_3a_4a_5$, the parity-check rules are $a_3 = a_2 \dotplus a_4$, $a_5 = a_1 \dotplus a_2 \dotplus a_4$.

The parity-check rules of a given (n,k) alphabet can be used to reduce greatly the complexity of instrumenting a maximum-likelihood detection scheme for use with the alphabet. The check places of the alphabet are numbered from 1 to $n - k$. The check places of each received (possibly erroneous) letter are examined in order and each check place is compared with the value predicted for that check place by the parity-check rules. A "0" is written if the numbers agree; a "1" is written otherwise. The resultant $(n - k)$-place binary sequence is called the "parity-check sequence" of the received letter.

It can be shown that all sequences in the jth row of Fig. 14.1 have the same parity-check sequence as S_j, $j = 1, 2, \ldots, 2^{n-k}$, $(S_1 = I)$. Furthermore, no two S's have the same parity-check sequence. A code book of 2^{n-k} entries can therefore be constructed which associates each S with its parity-check sequence. For each received sequence, the parity-check sequence is formed. The corresponding S is then found in the code book and is added (mod 2) to the received sequence. The resultant sequence will be a letter of the (n,k) alphabet and is the maximum-likelihood estimate of the transmitted letter.

14.10. GENERALIZED HAMMING ALPHABETS

If n and k are chosen so that $n + \binom{n-k}{2} + \binom{n-k}{3} \geq 2^{n-k} - 1$ a simple (n,k) alphabet and maximum-likelihood detector can be described which cannot be excelled by an alphabet of 2^k letters using n-digit binary sequences. That is, no other alphabet of the same size has a smaller average probability of error per letter. Values of n and k satisfying this inequality are given in Fig. 14.2.

n	k
2, 3, 4	1, 2, . . . , n
5, 6, . . . , 10	n-4, n-3, . . . , n
11, 12, . . . , 27	n-5, n-4, . . . , n
28, 29, . . . , 70	n-6, n-5, . . . , n
71, 72, . . . , 170	n-7, n-6, . . . , n
171, 172, . . . , 390	n-8, n-7, . . . , n

FIG. 14.2

Let $n - k = m$. Label the information positions by the integers from 1 to k and label the check positions by the integer from 1 to m. The parity-check rules for the alphabet can be constructed from a $k \times m$ matrix as shown in Fig. 14.3. The rows of the matrix are labeled by the information-position labels, the columns by the check-digit labels. The first row of the table contains m ones. Successive rows of the matrix are obtained by writing down in any order all the $\binom{m}{1}$ m-place binary sequences with exactly one 0, then in any order all the $\binom{m}{2}$ m-place binary sequences with exactly two 0's, etc., until the $k \times m$ matrix is complete. Let the element in the ith row and jth column of this matrix be c_{ij}. Let $a_1a_2 \cdots a_k$ be the binary sequence in the information positions of a letter of the alphabet. Then the jth check digit b_j is given by the parity-check rule

$$b_j = \sum_{i=1}^{k} c_{ij}a_i$$

where the summation is modulo 2 and $j = 1, 2, \ldots, m$.

The matrix of Fig. 14.3 also serves as a code book for the maximum-likelihood detector for this alphabet. For each received (possibly erroneous) letter, the parity-check symbol is formed as described in Sec. 14.9. That is, the binary sequence $f_1f_2 \cdots f_m$ is found where

$$f_j = e_j + \sum_{i=1}^{k} c_{ij}d_i \qquad j = 1, 2, \ldots, m$$

and where d_i is the binary symbol in the ith information position of the received letter and e_j is the binary symbol in the jth check position of the received letter. If the

	1	2	3	4	· · · ·	m
1	1	1	1	1	· · · ·	1
2	0	1	1	1	· · · ·	1
3	1	0	1	1	· · · ·	1
·	·	·	·	·	· · · ·	·
·	·	·	·	·	· · · ·	·
·	·	·	·	·	· · · ·	·
k	·	·	·	·	· · · ·	·

FIG. 14.3

resultant parity-check symbol $f_1f_2 \cdots f_m$ is the same as the rth row of the matrix of Fig. 14.3, the binary digit in the rth information place of the received letter should be altered. If the parity-check symbol for the received letter is not listed among the rows of the matrix and does not contain exactly three 1's, the binary digits in those check places of the received letter should be changed that correspond to the places of $f_1f_2 \cdots f_m$ that have the binary digit 1. If the parity-check symbol for the received letter contains exactly three 1's and does not appear as a row of the matrix of Fig. 14.3, a row of this matrix, say the ith, is located that has exactly four 1's, three of which coincide in position with the three 1's of the parity-check symbol. Let the fourth 1 in the ith row of the matrix be in column j. Then in the received letter the ith information place and the ith check place should be altered.

The alphabets and detector just described correct $2^{n-k} - 1$ single errors if $n \geq 2^{n-k} - 1$. If $2^{n-k} - 1 - \binom{n-k}{2} - \binom{n-k}{3} \leq n < 2^{n-k} - 1$, the alphabets correct all $\binom{n}{1}$ single errors and in addition $2^{n-k} - 1 - n$ double errors.

A special case of the foregoing alphabets occurs when n and k have the special values $n = 2^r - 1$, $k = n - r$, $r = 1, 2, 3, \ldots$. This was the case originally treated by Hamming,[13] and the above alphabets in this case are Hamming single-error-correcting alphabets when the information and check places are suitably chosen. In the original Hamming alphabet for $n = 2^r - 1$, $k = n - r$, the places of each letter are numbered from 1 to n. Places 1, 2, 4, 8, . . . , 2^{r-1} are the check places. Unlike the generalized alphabets described above the binary symbol to occupy a given check place is

Check Positions	*Positions Checked*
1	1, 3, 5, 7, 9, 11, 13, 15, 17, . . .
2	2, 3, 6, 7, 10, 11, 14, 15, 18, . . .
4	4, 5, 6, 7, 12, 13, 14, 15, 20, . . .
8	8, 9, 10, 11, 12, 13, 14, 15, 24, . . .
·	·
·	·
·	·

FIG. 14.4

determined as a linear (mod 2) sum of the entries in certain *check places* as well as of the entries in certain information places. The entry in place 2^j, $j = 0, 1 \ldots r-1$, is the mod 2 sum of the entries in all places i such that the usual binary expansion of the integer i has a 1 in the $(j+1)$st place from the right. Figure 14.4 shows the positions checked by the first few check places. Maximum-likelihood detection again involves the formation of parity-check sequences. When the parity-check sequence is written in reverse order (i.e., the place corresponding to check place 2^{r-1} is written leftmost, . . . the place corresponding to check place 1 is written rightmost) the resultant sequence is the binary expansion of an integer l from the set 0, 1, 2, . . . , n. If $l > 0$, the entry in the lth place of the received letter should be altered: if $l = 0$, all places should be left unaltered.

14.11. OTHER GROUP ALPHABETS

There are a number of other values of n and k for which it is known that group alphabets exist which cannot be excelled by any other alphabet of the same size. In most of these cases, there does not seem to be any very simple rule that associates the S's of the maximum-likelihood detector with the parity-check sequences. In Fig. 14.5, parity-check rules for these alphabets are given. The determination of S's and their parity-check sequences can be carried out as indicated in Sec. 14.9. In each case, the

$n = 8$	$n = 9$	$n = 10$	$n = 11$	$n = n$
$k = 2$	$k = 4$	$k = 5$	$k = 4$	$k = 1$
3 1	5 1 3 4	6 1 3 4 5	5 1 3	2 1
4 1	6 1 2 4	7 1 2 4 5	6 2 4	3 1
5 2	7 1 2 3	8 1 2 3 5	7 1 4	4 1
6 2	8 1 2 3	9 1 2 3 4	8 2 3	·
7 1 2	9 1 2 3	10 1 2 3 4 5	9 1 3 4	·
8 1 2			10 2 3 4	·
			11 1 2 3 4	n 1

FIG. 14.5

$n = 7$	$n = 8$	$n = 9$	$n = 9$	$n = 10$	$n = 10$	$n = 10$
$k = 2$	$k = 3$	$k = 2$	$k = 3$	$k = 2$	$k = 3$	$k = 4$
3 1	4 1	3 1	4 1	3 1	4 1	5 3 4
4 1	5 1 2	4 1	5 2	4 1	5 2	6 1 2 3
5 1	6 1 3	5 1	6 1 2	5 1	6 3	7 1 2 4
6 1 2	7 2 3	6 2	7 1 3	6 2	7 1 2	8 1 3 4
7 2	8 1 2 3	7 2	8 2 3	7 2	8 1 3	9 2 3 4
		8 1 2	9 1 2 3	8 1 2	9 2 3	10 1 2 3 4
		9 1 2		9 1 2	10 1 2 3	
				10 1 2		

$n = 11$	$n = 11$	$n = 12$	$n = 12$
$k = 3$	$k = 3$	$k = 2$	$k = 3$
3 1	4 3	3 1	4 1
4 1	5 3	4 1	5 2
5 1	6 2	5 1	6 3
6 2	7 1 3	6 1	7 1 2
7 2	8 1 3	7 2	8 1 2
8 2	9 1 2	8 2	9 1 3
9 1 2	10 1 2 3	9 2	10 2 3
10 1 2	11 1 2 3	10 1 2	11 1 2 3
11 1 2		11 1 2	12 1 2 3
		12 1 2	

FIG. 14.6

numbers refer to the places of a letter of the alphabet. The first k places are taken to be the information places. The numbers listed in a row in Fig. 14.5 give the numbers of the places whose entries must sum to zero modulo 2. For example, the parity-check rules for the (9,4) alphabet listed are: the entry in the 5th place is the sum mod 2 of the entries in the 1st, 3d, and 4th places; the entry in the 6th place is the sum mod 2 of the entries in the 1st, 2d, and 4th places; etc.

The (11,4) alphabet listed corrects all single and all double errors as well as 61 triple errors. It may be useful in a computer representing decimal digits in binary form since it has $2^4 = 16 > 10$ letters.

The alphabets whose parity-check rules are given in Fig. 14.6, while not known to be unexcelled by any other alphabet of the same size and n value, are known to be unexcelled by any (n,k) alphabet when the p_j of Sec. 14.8 are given by $p_j = p^j(1-p)^{n-j}$, $j = 0, 1, 2, \ldots, n$, $0 < p < \frac{1}{2}$. This is the case when transmitted binary digits are perturbed independently and have probability p of being altered. The maximum-likelihood detector for each alphabet can be constructed as in Sec. 14.9.

14.12. REED-MULLER ALPHABETS

The Reed-Muller alphabets are particular (n,k) alphabets for which $n = 2^m$, $k = \sum_{i=0}^{r} \binom{m}{i}$. For each integer value of m and $r \leq m$ there is one Reed-Muller alphabet. It is convenient to label the entries in the k information places of a typical letter of the alphabet by the symbols $a_0, a_1, a_2, \ldots, a_m, a_{12}, a_{13}, \ldots, a_{(m-1)m}, a_{123}, \ldots, a_{123\cdots r}, \ldots, a_{(m-r+1)(m-r+2)\cdots m}$. That is, the a's are labeled by all $k = \sum_{0}^{r} \binom{m}{i}$ combinations of the integers 1, 2, . . . , m taken r or fewer at a time. Similarly, it is convenient to label the entries in the check places by $b_{12\cdots r(r+1)}$, $b_{12\cdots r(r+2)}, \ldots, b_{12\cdots m}$. That is, the b's are labeled by all $2^m - k$ combinations of the integers 1, 2, . . . , m taken $r+1$ or more at a time. It will be convenient to denote a typical a or b with j subscript by $a_\alpha{}^{(j)}$ or $b_\alpha{}^{(j)}$, respectively. Let $C_{ij}{}^r$ be 1 or 0 according as $\binom{j-i-1}{r-i}$ is odd or even, respectively. The parity-check rules for the Reed-Muller alphabets can then be written

$$b_\alpha{}^{(j)} = \sum_{i=0}^{r} C_{ij}{}^r \sum_{\beta}{}' a_\beta{}^{(i)} \tag{14.12.1}$$

$j = r+1, r+2, \ldots, m$, all possible α. Here all sums are modulo 2 and the β sum is over all indices β that are i-fold combinations of the j integers that comprise the subscript α. For example, if $\alpha = 1346$, the β sum above is, for $i = 2$,

$$\sum_{\beta}{}' a_\beta{}^{(2)} = a_{13} \dotplus a_{14} \dotplus a_{16} \dotplus a_{34} \dotplus a_{36} \dotplus a_{46}$$

A maximum-likelihood detector for the Reed-Muller alphabets can be constructed as in Sec. 14.9. The construction of such a detector for large m and r is, however, laborious.

The Reed-Muller alphabets possess the following important feature: In the alphabet with parameters m and r, all pairs of letters are separated by a distance of at least 2^{m-r}. It is easily shown that this implies for these alphabets that among the S's of Fig. 14.1 will be found all n-digit binary sequences having $2^{m-r-1} - 1$ or fewer ones. A detector can be made for the alphabet by writing down the code book that lists these S's with their corresponding parity-check sequences. These latter are readily found from the

parity-check rules (14.12.1) above. For each received letter the parity-check sequence is computed. If the sequence is in the code book, the corresponding S is added modulo 2 to the received letter to form the corrected letter. If the parity-check sequence is not in the code book, no decision is made. This detector will correct all $(2^{m-r-1} - 1)$-fold or fewer errors in transmission. When more than $2^{m-r-1} - 1$ errors are made in transmission, the detector makes no decision.

The description of the Reed-Muller alphabets given here is considerably different from that given in Ref. 27. The detector described in the preceding paragraph gives performance equivalent to that described in Ref. 28 and is felt to be much simpler.

14.13. BOUNDED-DISTANCE DETECTORS

When all letters of a signaling alphabet are at least distance $2d + 1$ apart (d some positive integer) a simple detector that corrects up to all d-fold errors can be made by associating with each letter all vertices of the n cube that are distance d or less from the letter. In general there will be vertices of the cube which are more than d units distant from every letter of the alphabet. The detector makes no decision as to what letter was sent when such vertices are received sequences. Such detectors are known as "bounded-distance" detectors. An example was given in the last section.

A bounded-distance detector in general gives poorer performance than a maximum-likelihood detector. The justification for their use lies in the fact that they are generally of a simpler nature than the maximum-likelihood detector.

The collection of vertices distant d or less from a given vertex of the n cube is called a "sphere" of radius d with center at the given vertex. There are $\sum_{i=0}^{d} \binom{n}{i}$ points in such a sphere. A necessary condition that there exist a K-letter n-digit alphabet with all points at least distance $2d + 1$ apart is clearly $K \sum_{0}^{d} \binom{n}{i} \leq 2^n$.

For certain values of n and d, it is possible to find $K = 2^n \Big/ \sum_{0}^{d} \binom{n}{i}$ disjoint spheres of radius d that contain all 2^n vertices of the cube. Such an arrangement of spheres is said to be "close-packed." The centers of the spheres can be taken as letters of a signaling alphabet which allows correction of d-fold or fewer errors. The bounded-distance detector and maximum-likelihood detector for such "close-packed" alphabets coincide. The Hamming single-error-correcting alphabets are of this type. Only one other close-packed alphabet is known. It has 2^{12} letters, $n = 23$, $d = 3$.

14.14. PLOTKIN ALPHABETS

M. Plotkin[19] has worked out much detailed information about the quantity $A(n,l)$, the maximum number of letters possible in an n-digit alphabet with all points at least distance l apart. He also gives explicit rules for constructing n-digit alphabets of $K = 8m$ letters with minimum letter separation $2m$. Here $n = 4m$, and m can take any positive-integral value such that $4m - 1$ is a prime. These alphabets are not in general group alphabets. No simple detection rule (such as by means of parity checks) is known for them. Details of their construction can be found in Ref. 19.

14.15. THE BALSER-SILVERMAN-WAGNER CHANNEL AND DETECTOR

Balser and Silverman[27] compare the performance obtained on using various group alphabets for the transmission of binary data under circumstances in which the errors are introduced in a manner quite different from that described in Sec. 14.8. The detection scheme used is due to C. A. Wagner.

Balser and Silverman consider the transmission of binary messages over a band-limited electrical channel. A 0 is transmitted as a particular signal x_1 of duration T: a 1 is transmitted as a signal x_2 of duration T. In a period of duration T, a signal y is observed at the receiver. From the given noise properties of the channel, it is possible to compute $p(x_i|y)$, the conditional probability that x_i was transmitted knowing that y was received, $i = 1, 2$. If y is received and $p(x_1|y) \geq p(x_2|y)$ the receiver tentatively registers a 0; otherwise a 1. The quantities $\Delta p = |p(x_1|y) - p(x_2|y)|$ are computed for each received y and stored for use in correcting digits when the entire n-digit letter has been received.

In the simplest case treated by Balser and Silverman, an $(n,n-1)$ alphabet is used having the parity-check rule $b_1 = \sum_{1}^{n-1} a_i$ where b_1 is the check digit and the a's are the information digits. This parity rule is applied to the tentative sequence produced by the receiver. If the check is satisfied the tentative sequence is accepted. If the check fails, that digit of the tentative sequence is altered for which Δp is smallest. When more complicated (n,k) alphabets are used for the initial binary data, the Wagner detector uses the stored Δp in a suitably modified manner.

Reference 27 compares the performance of several different group alphabets both when the detector utilizes and when it ignores the stored Δp information. All calculations are performed for signals x_1 and x_2 of equal energy and under the assumption of additive white Gaussian noise. Comparisons are made for constant-input data rate, i.e., for constant rate of information digits. When the Δp information is not utilized, detection is by means of bounded-distance detectors rather than maximum-likelihood detection. Details of the results found are too lengthy for inclusion here. Generally speaking, utilization of the Δp information leads to a decrease in error probability. The effect is most pronounced when used on alphabets of small size.

The general problem of transmitting binary information over a band-limited electrical channel in the presence of additive white Gaussian noise has been much studied in information theory. More details on this channel and the problem of choosing good alphabets for it can be found in Ref. 27.

14.16. THE PROPOSITIONAL CALCULUS*

The fundamental logical and switching circuits used in digital computers are usually devices whose input and output wires have two significant states (e.g., a high and a low voltage) and whose output is a logical function of the inputs (e.g., the output voltage is high just in case *either* one input *or* the other input is high). The synthesis of complex circuits out of these building blocks is greatly facilitated by the use of the propositional calculus, the two-valued logic of the connectives "either . . . or," "and," "not," etc.

The calculus contains the following symbols: the propositional *variables* "p," "q," "r," "s," etc.; the truth *values* "true" ("1," "t") and "false" ("0," "f"); and an indefinite list of logical *connectives*, of which the following are the most common:

Negation	Not-p	$\bar{p}$, $\sim p$, p'
Conjunction	p and q	$p \cdot q$, pq, $p \,\&\, q$
Inclusive disjunction	p or q (or both)	$p \vee q$, $p + q$
Conditional implication	If p then q	$p \supset q$
Biconditional, equivalence	p if and only if q	$p \equiv q$
Exclusive disjunction, inequivalence	p or q but not both	$p \not\equiv q$
Stroke	Not both p and q	$p \mid q$
Dual stroke	Neither p nor q	$p \downarrow q$

Meaningful (well-formed) formulas may be constructed by compounding these symbols with parentheses in an intuitively meaningful way; e.g., "$(p\bar{q}) \vee (\bar{p}q)$" is an

* Sections 14.16 to 14.26 are by Arthur W. Burks.

alternative way of expressing exclusive disjunction. Extra parentheses can often be dropped by the convention that a negation is weaker than a conjunction and a conjunction weaker than a disjunction; e.g., "$p \vee \sim qr$" means "$[p \vee ((\sim q) \cdot r)]$" and not "$[p \vee \sim(q \cdot r)]$" or "$[(p \vee \sim q) \cdot r]$." All the logical connectives can be expressed in terms of negation and disjunction; or in terms of negation and conjunction; or in terms of the stroke alone; or in terms of the dual stroke alone.

The application of the calculus may be illustrated in connection with the conjunction element (gate) of Fig. 14.7. Each input wire is labeled with a variable which is interpreted to mean that the labeled wire is activated (high); e.g., "p" is true if the wire labeled "p" is high, otherwise "p" is false. The output wire is labeled with the corresponding logical function ("$p \cdot q$") or with another variable ("r") which is set equal to this function. This is the *normal interpretation;* the *dual interpretation,* in which "true" is associated with a low voltage, may also be used, but confusion results from mixing the two interpretations.

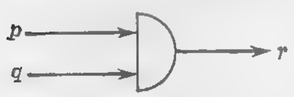


FIG. 14.7. $r = (p \cdot q)$.

14.17. BOOLEAN ALGEBRA

The calculus just described is a Boolean algebra. Sometimes the class interpretation of Boolean algebra is used in computer work. On this interpretation the symbols "$\cup$" (cup) and "$\cap$" (cap) represent the operation of class union (sum) and class intersection (product), respectively. In so far as these symbols are synonymous with disjunction and conjunction they are just alternative notations, but the interpretation in terms of class operations is not so simple as the one given here.

The logical connectives may be defined arithmetically, e.g.,

$$\begin{aligned}(p \vee q) &= (p + q - (p \times q))\\ (p \cdot q) &= (p \times q)\\ \bar{p} &= (1 - p)\end{aligned}$$

It is usually more convenient to work with the logical symbols directly, but the arithmetical representation is sometimes useful for solving logical problems on digital computers.

14.18. TRUTH TABLES

Of great importance for computer work is the idea of a formula which is true for all possible values of the variables; such a formula is called a *valid formula* or *tautology.*

A formula may be tested for validity by evaluating it for all possible values of its variables; e.g., "$p \vee \bar{p}$" is a tautology since both "$0 \vee \bar{0}$" and "$1 \vee \bar{1}$" are true. This evaluation may be carried out systematically by means of a *truth table.* Thus the formula at the head of the rightmost column below is a tautology because that column consists entirely of 1's. Every other column has at least one 0 in it; so no other formula heading a column is a tautology.

	p	q	$p \cdot q$	$\overline{(pq)}$	$\bar{p}$	$\bar{q}$	$\bar{p} \vee \bar{q}$	$\overline{(pq)} \equiv (\bar{p} \vee \bar{q})$
$\bar{p}\bar{q}$	0	0	0	1	1	1	1	1
$\bar{p}q$	0	1	0	1	1	0	1	1
$p\bar{q}$	1	0	0	1	0	1	1	1
pq	1	1	1	0	0	0	0	1

The columns headed by the two variables ("p" and "q") contain all the 2-bit binary numbers, and hence in general an n-variable truth table contains 2^n rows.

Each row is headed by a specific n-bit binary number giving the particular assignment of truth values to the variables for that row. This assignment can also be represented by a conjunction of all the variables with the appropriate ones (if any) negated; see the leftmost columns in the above table. We shall call such a conjunction a *basic conjunction.*

14.19. DISJUNCTIVE NORMAL FORMS

A function which is false for all values of the variables is a *contradiction;* e.g., "$p \cdot \bar{p}$." It is clear from an inspection of a truth table that every logical function which is not a contradiction is equivalent to a disjunction of those basic conjunctions of its variables for which it is true; e.g., "$\overline{(pq)}$" is equivalent to "$\bar{p}\bar{q} \vee \bar{p}q \vee p\bar{q}$." This form is called the (*expanded*) *disjunctive normal form.*

The disjunctive normal form expression of a tautology of n variables will contain all 2^n basic conjunctions of those n variables, and so one method of testing for validity is to expand (by using other tautologies) a formula into disjunctive normal form to see if this expansion contains all the basic conjunctions.

For a specific ordering of the basic conjunctions a disjunctive normal form may be expressed more briefly by writing the corresponding column of the truth table; e.g., "$\overline{(pq)}$" may be expressed by the binary number 1110. For a particular ordering of the variables the disjunctive normal form may be expressed arithmetically by using the binary numbers which head the rows of the truth table in place of the corresponding basic conjunctions; e.g., "$\overline{(pq)}$" is expressed by $00 \vee 01 \vee 10$.

The following is an important consequence of the fact that every logical function may be expressed in disjunctive normal form. Suppose it is desired to realize an arbitrary logical expression A and that there are available wires representing not only the variables of A but also the negations of these variables; the latter would be true, for example, if the desired circuit were driven by symmetrical flip-flops, or if the conjunction elements have inhibitory as well as activating inputs. Under these circumstances any logical function A may in principle be exemplified in a circuit consisting of conjunctive elements (one for each basic conjunction in the disjunctive normal form expansion of A) driving a disjunctive element (with as many inputs as there are basic conjunctions in the disjunctive normal form expression of A). It should be noted that such a circuit has a *depth* of two elements; i.e., there are two logical elements between each input and the output. Circuits constructed by the use of additional logical elements and with a depth greater than 2 may, of course, be more economical.

14.20. TESTING FOR VALIDITY

It has already been mentioned that one can test a formula for validity by constructing a truth table for that formula. This procedure is purely routine and hence can readily be mechanized. This can be done by connecting up a circuit which realizes the formula under investigation, and testing the output of this circuit for all possible inputs; or by programming the problem on a general-purpose digital computer; or by means of a logic machine which stores the formula and determines its truth value for every possible assignment of truth values to the variables.[4]

The truth-table method of testing validity is tedious, and impractical for expressions with a large number of variables; e.g., for six variables 2^6 or 64 rows are needed. A procedure which relies on ingenuity is generally quicker for the human being (as contrasted to a machine). Simple formulas may be tested intuitively, and a list of simple tautologies such as the following is easily remembered after a little practice:

Double negation	$p \equiv \bar{\bar{p}}$	
Excluded middle	$p \vee \bar{p}$	$(p \vee \bar{p}) \equiv 1$
Contradiction	$(p\bar{p}) \equiv 0$	
Commutation	$(p \cdot q) \equiv (q \cdot p)$	$(p \vee q) \equiv (q \vee p)$

Association	$[p \cdot (qr)] \equiv [(pq) \cdot r]$
	$[p \vee (q \vee r)] \equiv [(p \vee q) \vee r]$
Distribution	$[p \cdot (q \vee r)] \equiv [pq \vee pr]$
	$[p \vee qr] \equiv [(p \vee q) \cdot (p \vee r)]$
DeMorgan's theorem	$\overline{(pq)} \equiv (\bar{p} \vee \bar{q})$
	$\overline{(p \vee q)} \equiv (\bar{p}\bar{q})$
Absorption	$(p \vee 1) \equiv 1 \qquad (p \cdot 1) \equiv p$
	$(p \vee 0) \equiv p \qquad (p \cdot 0) \equiv 0$
	$(p \vee p) \equiv p \qquad (p \cdot p) \equiv p$
	$p \equiv (p \vee pq) \qquad p \equiv [p(p \vee q)]$
	$(pq \vee \bar{p}r) \equiv (pq \vee \bar{p}r \vee qr)$
Simplification	$(p \vee q) \equiv (p \vee \bar{p}q)$
	$(pq) \equiv [p \cdot (\bar{p} \vee q)]$
Multiplication	$[(p \vee q) \cdot (r \vee s)] \equiv (pr \vee ps \vee qr \vee qs)$

With a working stock of simple tautologies one can establish other tautologies by using the following principles.

Interchange of Equivalents. If $A \equiv B$ is a tautology, and D results from C by replacing A by B in C, then $C \equiv D$ is a tautology; e.g., by the use of DeMorgan's theorem we can establish that $[p \cdot \overline{(qr)}] \equiv [p \cdot (\bar{q} \vee \bar{r})]$.

Principle of Substitution. The result of substituting any well-formed formula for a variable of a tautology is a tautology; e.g., the result of substituting "rs" for "p" in the first form of the law of excluded middle is the tautology "$rs \vee \overline{(rs)}$."

Modus ponens. If $A \supset B$ is a tautology, and A is a tautology, then B is a tautology; e.g., "$p \supset (q \supset p)$" is a tautology and so is "$r \vee \bar{r}$," and by substituting the latter for "p" and using modus ponens we derive the tautology "$q \supset (r \vee \bar{r})$." For a fuller treatment of these methods see any standard textbook in symbolic logic.[8,9,21]

14.21. DUALITY

Another principle useful for constructing tautologies is a generalization of DeMorgan's theorem called duality. The dual A^d of a formula A is formed by interchanging the following symbols in pairs:

0	1
p	$\sim p$ (and the same for every variable)
$\sim$	$\sim$
$\vee$	$\cdot$
$\equiv$	$\not\equiv$
$\downarrow$	$\mid$

The *principle of duality* states that the following are tautologies:

$$\bar{A} \equiv A^d \qquad A \equiv \bar{A}^d$$

Hence if $A \equiv B$ is a tautology so is $A^d \equiv B^d$ and if A is a tautology so is $\bar{A}^d$.

It is worth noting that a shift from the *normal* to the *dual* interpretation of the propositional calculus constitutes an interchange of each variable with its dual, and hence results in an interchange of dual connectives. Thus applying duality to

$$r \text{ is high} \equiv (s \text{ is high } \textit{or } t \text{ is high})$$

we get

$$r \text{ is low} \equiv (s \text{ is low } \textit{and } t \text{ is low})$$

since "$\sim(p \not\equiv q) \equiv (p \equiv q)$" is a tautology. This sometimes is expressed by saying that a "positive or" is a "negative and."

The disjunctive normal form expression of the formula $\bar{A}$ is a disjunction of the basic conjunctions for which A is false. The dual $\bar{A}^d$ is a conjunction of disjunctions and states the conditions under which A is true; it is called the (*expanded*) *conjunctive normal form* of A. Thus, if A is "$pq \vee p\bar{q}$," $\bar{A}$ is "$p\bar{q} \vee pq$," $\bar{A}^d$ (and hence the

conjunctive normal form of A) is "$(p \vee q) \cdot (\bar{p} \vee \bar{q})$." The conjunctive normal form has the same utility as the disjunctive normal form.

14.22. USING LOGIC TO DESIGN CIRCUITS

A common design problem is that of constructing a circuit to realize a given formula. For example, the sum digit s_i and carry-out digit c_{i-1} are the following functions of the addend digit a_i, augend digit b_i, and carry-in digit c_i:

$$s_i = (a_i \not\equiv b_i \not\equiv c_i) \tag{14.22.1}$$

$$c_{i-1} = (a_ib_i \vee a_ic_i \vee b_ic_i) \tag{14.22.2}$$

and the problem may be to construct a net to perform these operations. If there are circuit building blocks available corresponding to the logical connectives used in the formulas then the solution consists of a straightforward interconnection of these elements according to the pattern of the formulas; see Fig. 14.8 for a net realizing c_{i-1}, where the rightmost element is a disjunction (buffer) circuit.

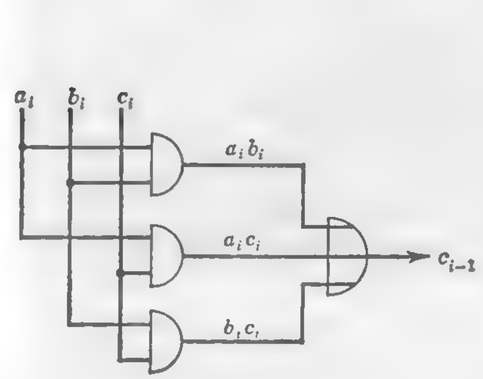


FIG. 14.8. Carry circuit.

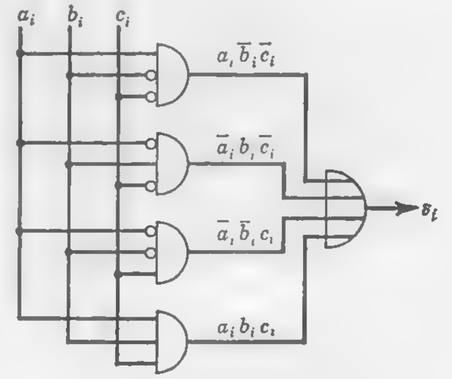


FIG. 14.9. Sum circuit.

There may not, however, be circuit components which directly realize the connectives used in the formula. Thus we may wish to construct a circuit for (14.22.1) out of conjunction and disjunction elements, where the conjunction elements may have inhibiting inputs (represented by small circles, see Fig. 14.9) as well as activating inputs. In this case the given formula is transformed by any of the afore-mentioned techniques into an equivalent one containing only the allowed connectives. Thus it may be shown that the disjunctive normal form of s_i is

$$a_i\bar{b}_i\bar{c}_i \vee \bar{a}_ib_i\bar{c}_i \vee \bar{a}_i\bar{b}_ic_i \vee a_ib_ic_i \tag{14.22.3}$$

either by the use of the tautology

$$(p \not\equiv q) \equiv (p\bar{q} \vee \bar{p}q) \tag{14.22.4}$$

together with the tautologies previously cited, or by proof of the fact that "$p_1 \not\equiv p_2 \not\equiv p_3 \cdots \not\equiv p_n$" is true just in case an odd number of the variables are true.

Whenever there is a one-to-one correspondence between logical connectives and available circuit elements it is easy to pass from formula to circuit and vice versa. Indeed, formula (14.22.3) and Fig. 14.9 say essentially the same thing, one in logical symbols, the other diagrammatically. We shall speak of such transformations from logical formulas to diagrams as *direct translations*.

Another design problem is the following: Given a net, does it realize a specified formula; e.g., does Fig. 14.10 realize s_i and c_{i-1}? A related problem is that of determining whether two nets realize the same formula. Because of the ease with

which direct translations between formulas and net diagrams may be made, both these problems reduce to the question of whether two formulas A and B are equivalent, i.e., whether $A \equiv B$ is a tautology.

Thus a direct translation of Fig. 14.10 gives

$$p_i = [(a_i \vee b_i) \cdot \overline{(a_i b_i)}] \tag{14.22.5}$$

$$s_i = [(p_i \vee c_i) \cdot \overline{(p_i c_i)}] \tag{14.22.6}$$

$$c_{i-1} = (a_i b_i \vee p_i c_i) \tag{14.22.7}$$

When it is recognized that

$$(p \not\equiv q) \equiv [(p \vee q) \cdot \overline{(pq)}] \tag{14.22.8}$$

is a tautology (14.22.5) and (14.22.6) may be quickly reduced to (14.22.1); (14.22.7) may be shown equivalent to (14.22.2) by the use of (14.22.4) and other tautologies.

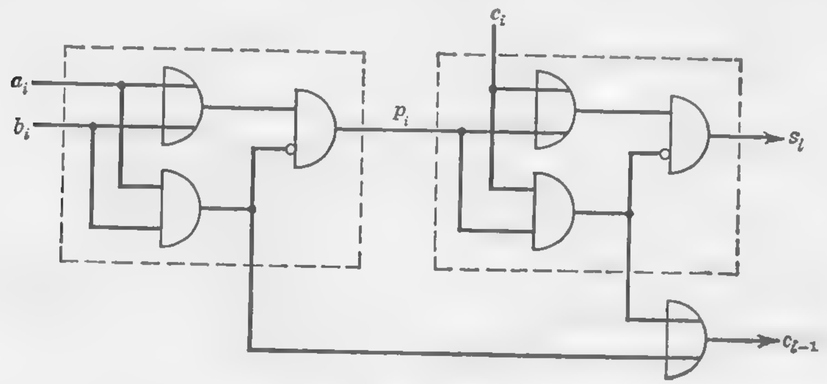


FIG. 14.10. Adder.

The nets within the dotted lines of Fig. 14.10 are called half-adders, since two in tandem give a full stage of a binary adder (which in turn may be used either sequentially or in parallel with $n - 1$ identical units to make a binary adder for an n-bit number). Formula (14.22.8) shows that with regard to its upper output a half-adder is an inequivalence ($\not\equiv$) element. It is worth noting that an inequivalence element is also a complementer: if wire a_i is inactive ("a_i" false) then $p_i \equiv b_i$, while if a_i is active ("a_i" true) then $p_i \equiv \bar{b}_i$.

14.23. MINIMALITY FOR A SINGLE FUNCTION

A large class of logical design problems arise out of the need of constructing the simplest net corresponding to a given set of functional expressions. If "simple" is given a logical definition (e.g., in terms of the number of elements, or in terms of the number of inputs to elements) these become problems in pure logic. For example, the problem of constructing a minimal net realizing the formula A is the logical problem of determining which of the formulas equivalent to A has this minimal property.

A special case of this last problem is of great interest because of the basic importance of disjunctive normal form. It arises in the following way: The formula "$pq \vee p\bar{q} \vee \bar{p}\bar{q}$" is equivalent to the formula "$p \vee \bar{p}\bar{q}$." The latter is a disjunction of conjunctions, each conjunction being composed of variables from the original formula or their negations. It is called a (*contracted*, as contrasted with expanded) *disjunctive normal form*. "$p \vee \bar{q}$" is a still simpler contracted disjunctive normal form of the original formula. The problem, roughly put, is to find the simplest disjunctive normal form expression of a given formula.

It often happens that some of the logically possible input conditions to a circuit never occur (e.g., in a binary-decimal conversion six of the 4-bit binary numbers may never occur), and in such cases it does not matter how the circuit behaves. The

minimality problem can then be broadened by saying that a circuit is required which realizes the given formula A in all the cases that do occur, and which is minimal. If the minimal circuit is a direct translation of B and the conditions which do occur are given by C, it is not required that $A \equiv B$ be a tautology but only that $C \supset (A \equiv B)$ is.

If the formula to be minimized is put into expanded disjunctive normal form it may be simplified by replacing two conjunctions by one in accordance with the tautology

$$p \equiv (pq \vee p\bar{q}) \tag{I}$$

where any formula can be substituted for "p" and the order of the variables is immaterial, e.g.,

$$abd \equiv (abcd \vee ab\bar{c}d)$$

Similarly it may be possible to replace four, eight, sixteen, . . . conjunctions by use of the tautologies

$$p \equiv (pqr \vee pq\bar{r} \vee p\bar{q}r \vee p\bar{q}\bar{r})$$

$$p \equiv (pqrs \vee pqr\bar{s} \vee pq\bar{r}s \vee p\bar{q}rs \vee pq\bar{r}\bar{s} \vee p\bar{q}r\bar{s} \vee p\bar{q}\bar{r}s \vee p\bar{q}\bar{r}\bar{s})$$

etc.

respectively, but clearly these reductions can also be accomplished by repeated applications of (I). A common method of solving this particular minimality problem is to test systematically for all possible simplifications of this kind, and then make a minimal selection from the conjunctions that result. A nonoccurring case (or conjunction) may be used to find a simpler conjunction, but it need not be represented in the final disjunction. There are many ways of systematizing this simplification process, some of which use truth tables,[29] or variants of them,[15] and others do not.[22] The method illustrated next is an extension of Quine's method to cover nonoccurring cases.

The expression to be minimized is the disjunction of

$$\bar{a}bc \quad \checkmark \tag{14.23.1}$$

$$ab\bar{c} \quad \checkmark \tag{14.23.2}$$

$$\bar{a}\bar{b}c \quad \checkmark \tag{14.23.3}$$

with the following cases never occurring:

$$abc \quad \checkmark \tag{14.23.4}$$

$$\bar{a}\bar{b}\bar{c} \quad \checkmark \tag{14.23.5}$$

Starting with (14.23.1) we compare it successively with the terms following it until we can use the tautology "$p \equiv (pq \vee p\bar{q})$." Combining (14.23.1) with (14.23.3) and (14.23.4), respectively, we get

$$\bar{a}c \quad \text{from (14.23.1) and (14.23.3)} \tag{14.23.6}$$

$$bc \quad \text{from (14.23.1) and (14.23.4)} \tag{14.23.7}$$

A check is placed opposite (14.23.1) and (14.23.3) to show that they will not be in the minimal formula. (The nonoccurring cases will not be in the minimal formula; so they are checked at the outset.) Warning: Comparisons of (14.23.1) with the other formulas cannot be stopped after (14.23.6) is found because (14.23.1) may be useful again, as it was in finding (14.23.7). Applying the above procedure to (14.23.2) we get

$$ab \quad \text{from (14.23.2) and (14.23.4)} \tag{14.23.8}$$

and so we check (14.23.2); (14.23.3) and (14.23.5) give

$$\bar{a}\bar{b} \quad \text{from (14.23.3) and (14.23.5)} \tag{14.23.9}$$

(14.23.4) and (14.23.5) give no further simplification, nor does (14.23.6), (14.23.7), (14.23.8), or (14.23.9).

The desired minimal formula is a simplest disjunction of some or all of the unchecked formulas [these are (14.23.6), (14.23.7), (14.23.8), and (14.23.9) in our example] which is equivalent to the original formula [in our example this is the disjunction of (14.23.1), (14.23.2), and (14.23.3)]. If a basic conjunction of the original formula is unchecked (this case does not occur in our example) it will appear in the final formula, while if it is checked some unchecked formula derived from it must appear. Since (14.23.6) was derived from (14.23.1) and (14.23.3), while (14.23.8) was derived from (14.23.2), "$\bar{a}c \vee ab$" is the minimal formula in our example.

If the formula to be minimized is given in contracted disjunctive normal form it must be transformed into its expanded disjunctive normal form before the foregoing technique is applicable. Quine has devised a method in which this expansion is unnecessary.[20] It is a useful method but not always superior to the one just illustrated because in some cases the easiest way to find which of the unchecked formulas are to be used in the minimal expression involves expanding the given expression into expanded disjunctive normal form. As formulated, Quine's method is not applicable to problems with nonoccurring cases, but it may be extended to this type of problem by combining it with the method illustrated above (with some modifications of each).[15]

14.24. COMPLETE DECODING NETS

A complete decoding net is a net with input wires for n variables "p_i", . . . , "p_n" (and their negations) and with 2^n output wires, each realizing one of the 2^n basic conjunctions of these n variables. Such a circuit completely decodes the n-bit input word in the sense that for each word exactly one output wire is active; for example, it decodes a 12-bit address into 4,096 separate output wires, exactly one of which is activated for each different address. It follows from the fact that every logical function has an expanded disjunctive normal form that any function of the variables "p_i", . . . , "p_n" can be formed by disjoining (through a disjunction element) the appropriate outputs of the complete decoding net.

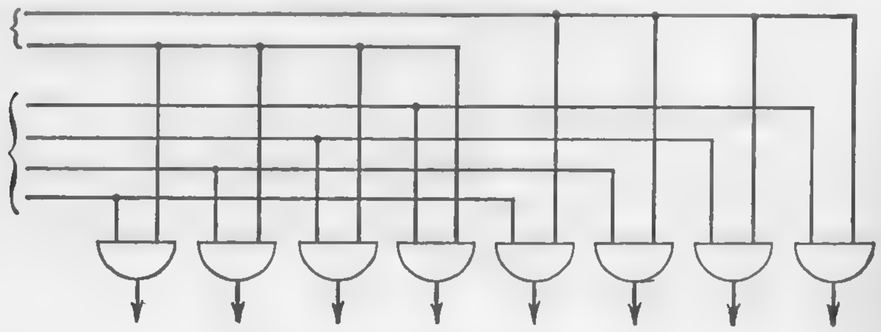


FIG. 14.11. Multiplicative switch.

There are a number of systematic ways of constructing complete decoding nets. All these use the multiplicative switch as a building block. Figure 14.11 shows a multiplicative switch with *bracketed input sets* of two and four wires, and Fig. 14.12 is a box abbreviating it. Figures 14.11 and 14.12 are of *dimensions* "2 by 4." A *multiplicative switch* has bracketed sets (dimensions) of $i_1, i_2, \ldots, i_n$ input wires ($i_1 = 2$, $i_2 = 4$ in Fig. 14.11) and $i_1 \times i_2 \times \ldots \times i_n$ conjunction elements ($i_1 \times i_2 = 8$ in Fig. 14.11) each with n input wires ($n = 2$ in Fig. 14.11). A given conjunction element is connected to exactly one wire of each bracketed set, and no two conjunction elements are connected in the same way.

An *exponential switch* for n variables is a multiplicative switch with n bracketed sets of two input wires each. See Fig. 14.13, which is a 2 by 2 by 2 switch.

A *tree* for n variables is a sequence of $n - 1$ multiplicative switches of dimensions (input sets) 2 by 2, 2 by 4, 2 by 8, . . . , 2 by 2^{n-1}, each switch driving its successor in the way illustrated in Fig. 14.14.

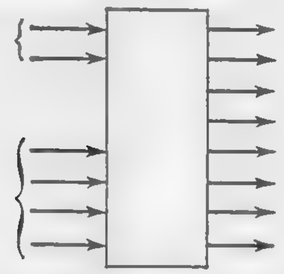


FIG. 14.12. Simplified representation of a multiplicative switch.

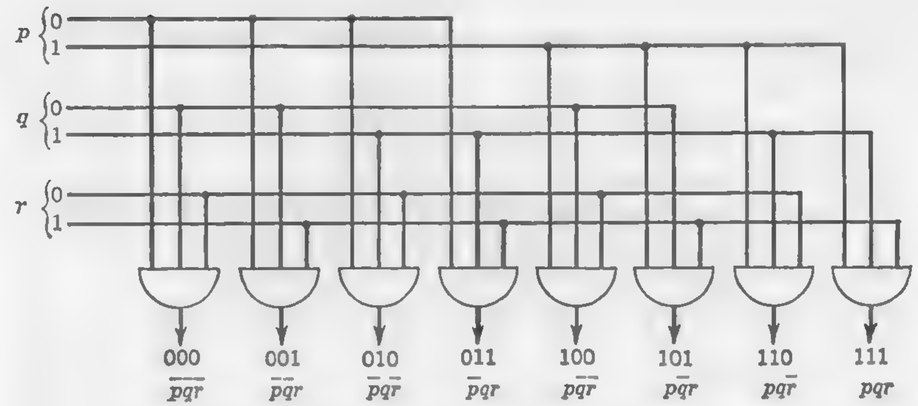


FIG. 14.13. Exponential switch.

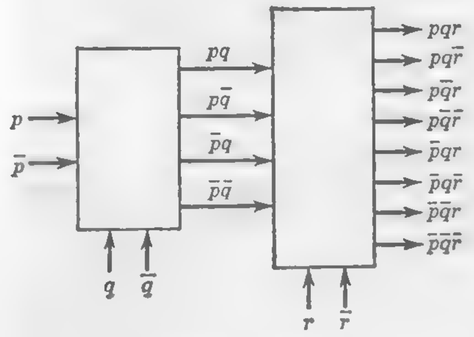


FIG. 14.14. Tree.

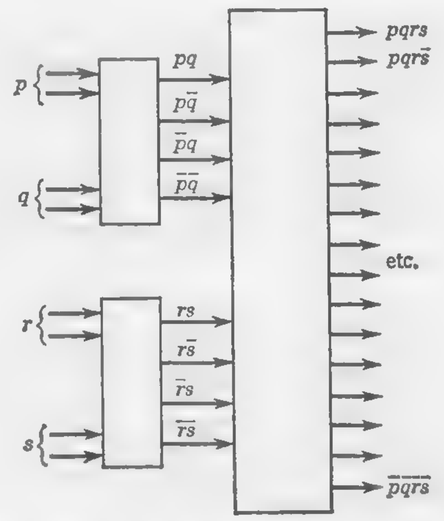


FIG. 14.15. Balanced multiplicative switch.

A *balanced multiplicative switch net* for four variables is illustrated in Fig. 14.15. A precise definition is complicated, but the construction procedure for a complete decoding net of n variables is roughly as follows: The final multiplicative switch has two bracketed input sets; if n is even these are each of $2^{n/2}$ input wires, while if n is odd they are of $2^{\frac{n+1}{2}}$ and $2^{\frac{n-1}{2}}$ input wires, respectively. This switch is in turn driven by two switches each of which is balanced (or nearly balanced) in a similar manner.

14.25. MINIMALITY FOR COMPLETE DECODING NETS

The engineer is often faced with the problem of finding a minimal circuit of more than one output. The solution of this problem is not in general the sum of the solutions of the constituent one-output problems since two or more outputs can use the same circuits in common; e.g., in Fig. 14.15 the pq input to the final switch is used in the four outputs $pqrs$, $pqr\bar{s}$, $pq\bar{r}s$, $pq\bar{r}\bar{s}$.

A useful economy criterion for complete decoding circuits is the *element-input count*, the number of (element) input wires of the net. This is 24 for Fig. 14.13, since there are 8 conjunction elements of 3 inputs each. For a standard way of constructing these nets out of crystal diodes the element-input count equals the number of diodes.[1]

The element-input count of exponential switches is given by $n2^n$ and that of trees by $2^{n+2} - 8$. The element-input count for small balanced multiplicative switch nets is given in the table below, and the count $C(n)$ for larger n can be derived from these by the recursion

$$C(n) = 2^{n+1} + C([n/2]) + C\left(\left[\frac{n+1}{2}\right]\right)$$

where "$[x]$" means "the integral part of x." It has been proved that the balanced multiplicative switch net is a minimal circuit out of all possible complete decoding circuits constructed from conjunction elements.[5]

No. of variables	Exponential switch	Tree	Balanced multiplicative switch net
2	8	8	8
3	24	24	24
4	64	56	48
5	160	120	96

Minimality is not the only utility criterion of a net. Thus an exponential switch may be preferred because it has an element depth of only one and hence may be faster than the balanced switch, or a tree may be preferred because it is especially adapted to the available circuit elements (e.g., relays). The load on the input wires of a tree may be balanced by a process called "folding."[6]

14.26. NETS WITH DELAYS

The propositional calculus must be extended for the analysis of circuits with memory or time delays. This may be done by adding to each variable (e.g., "f") a subscript "t" for time (giving "f_t"). "t" ranges over the nonnegative integers and "f_t" is true just in case the wire labeled "f" is active at time t.

A single additional element, the *delay element*, is needed to represent memory circuits.[7] Its operation is symbolized by the delay operator "D," which is defined recursively by (see Fig. 14.16)

$$f_t = Dg_t \qquad \begin{cases} f_0 = 0 \\ f_{t+1} = g_t \end{cases}$$

Other operators may be defined by means of this and the logical connectives. Thus operator "A," meaning "has been activated," is defined by

$$f_t = Ah_t = D(h_t \vee f_t)$$

i.e.,

$$f_0 = 0 \quad \text{and} \quad f_{t+1} = (h_t \vee f_t)$$

and the corresponding net is shown in Fig. 14.16.

Quantifiers may also be used. The behavior of f_t of Fig. 14.16 is given by the formula

$$f_t = (\exists x)(x < t \cdot h_x)$$

which means "f is energized at time t just in case there is a time x, earlier than t, at which h was activated."

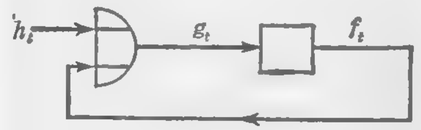


FIG. 14.16. Delay circuit.

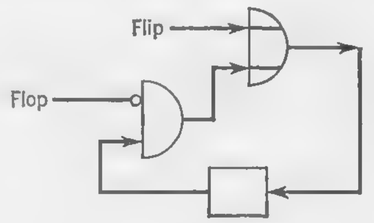


FIG. 14.17. Flip-flop.

Though the delay symbol is derived from delay-line memory devices, the logical behavior of static memory devices can also be represented by its use. Thus a flip-flop can be represented by the diagram of Fig. 14.17.

References

1. Brown, D. R., and N. Rochester: Rectifier Networks for Multi-position Switching, *Proc. IRE*, **37**:139–147 (February, 1949).
2. Burks, A. W.: The Logic of Fixed and Growing Automata, Proceedings of an International Symposium on the Theory of Switching, Part I, pp. 147–188, *Harvard Univ. Annals*, **29**, Harvard University Press, 1959.
3. Burks, A. W., and Hao Wang: The Logic of Automata, *J. Assoc. Computing Machinery*, **4**(2): 193–218 (April, 1957); **4**(3): 279–297 (July, 1957).
4. Burks, A. W., *et al.*: An Analysis of a Logical Machine Using Parenthesis-free Notation, *Mathematical Tables and Other Aids to Computation*, **8**:53–57 (April, 1954).
5. Burks, A. W., *et al.*: Complete Decoding Nets: General Theory and Minimality, *J. Soc. Ind. Appl. Math.*, **2**:201–243 (December, 1954).
6. Burks, A. W., *et al.*: The Folded Tree, *J. Franklin Inst.*, **260**:9–24 (July, 1955); 115–126 (August, 1955).
7. Burks, A. W., and Jesse B. Wright: Theory of Logical Nets, *Proc. IRE*, **41**:1357–1365 (October, 1953).
8. Copi, Irving M.: *Symbolic Logic*, The Macmillan Company, New York, 1954.
9. Copi, Irving M.: *Introduction to Logic*, The Macmillan Company, New York, 1953.
10. Elias, P.: Coding for Noisy Channels, *IRE Convention Record*, part 4, New York: 37–45, 1955.
11. Gilbert, E. N.: A Comparison of Signalling Alphabets, *Bell System Tech. J.*, **31**:504–522 (May, 1952).
12. Golay, M. J. E.: Correspondence, Notes on Digital Coding, *Proc. IRE*, **37**:657 (June, 1949). Binary Coding, *IRE Trans.*, 1954 Symposium on Information Theory, MIT, **PGIT-4**:23–28 (September, 1954).
13. Hamming, R. W.: Error Detecting and Error Correcting Codes, *Bell System Tech. J.*, **29**:147–160 (April, 1950).
14. Harris, Bernard: An Algorithm for Determining Minimal Representations of a Logic Function, *IRE Trans. on Electronic Computers*, **EC-6**(2): 103–107 (June, 1957).
15. Karnaugh, M.: The Map Method for Synthesis of Combinational Logic Circuits, *Trans. AIEE*, part 1, *Communication and Electronics*, **72**:593–599 (November, 1953).
16. McCluskey, E. J., Jr.: Minimization of Boolean Functions, *Bell System Tech. J.*, **35**:1417–1444 (November, 1956).
17. Mealy, G. H.: A Method for Synthesizing Sequential Circuits, *Bell System Tech. J.*, **34**:1045–1079 (September, 1955).
18. Moore, E. F.: Gedanken-Experiments on Sequential Machines, C. E. Shannon and J. McCarthy (eds.), *Automata Studies*, pp. 129–153, *Annals of Mathematics Studies*, **34**, Princeton University Press, Princeton, N.J., 1956.
19. Plotkin, M.: Binary Codes with Specified Minimum Distance, *Moore School Research Division Rept.* 51-20, University of Pennsylvania, Philadelphia, 1951.

20. Quine, W. V.: A Way to Simplify Truth Functions, *Am. Math. Monthly*, **62**:627–631 (November, 1955).
21. Quine, W. V.: *Methods of Logic*, Henry Holt and Company, Inc., New York, 1950.
22. Quine, W. V.: The Problem of Simplifying Truth Functions, *Am. Math. Monthly*, **59**:521–531 (October, 1952).
23. Reed, I. S.: A Class of Multiple-error-correcting Codes and the Decoding Scheme, *IRE Trans.*, 1954 Symposium on Information Theory, MIT, **PGIT-4**:38–49 (September, 1954).
24. Richards, R. K.: *Arithmetic Operations in Digital Computers*, D. Van Nostrand Company, Inc., Princeton, N.J., 1955.
25. Shannon, Claude: A Symbolic Analysis of Relay and Switching Circuits, *Trans. AIEE*, **57**:713–723 (1938).
26. Shapiro, H. S., and D. L. Slotnick: On the Mathematical Theory of Error-correcting Codes, *IBM Research Development*, **3**(1): 25–34 (January, 1959).
27. Silverman, R. A., and M. Balser: Coding for Constant-data-rate Systems, *IRE Trans.*, 1954 Symposium on Information Theory, MIT, **PGIT-4**:50–63 (September, 1954).
28. Slepian, D.: A Class of Binary Signalling Alphabets, *Bell System Tech. J.*, **35**:203–234 (January, 1956).
29. The Staff of the Harvard University Computation Laboratory: *Synthesis of Electronic Computing and Control Circuits, Annals*, **27**, Harvard University Press, Cambridge, Mass., 1951.
30. Urbano, R. H., and R. K. Mueller: A Topological Method for the Determination of the Minimal Forms of a Boolean Function, *IRE Trans. on Electronic Computers*, **EC-5**(3): 126–132 (September, 1956).

Section 15

DIGITAL-COMPUTER ARITHMETIC

By

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

CONTENTS

15.1. The Binary Number System. 15–2
15.2. The Octal Number System... 15–3
15.3. The Hexadecimal Number System..... 15–3
15.4. Ternary Number Systems... 15–4
15.5. Representation of Decimal Digits..... 15–4
15.6. Notation and Symbols..... 15–5
15.6.1. Flip-flop Notation..... 15–5
15.6.2. AND-gate Notation..... 15–5
15.6.3. OR-gate Notation..... 15–5
15.6.4. Inverters..... 15–5
15.6.5. Pulse Designations..... 15–6
15.6.6. Delay Lines..... 15–6
15.6.7. Timing Considerations... 15–6
15.7. Counters..... 15–6
15.7.1. A Decimal Counter..... 15–6
15.8. Serial and Parallel Handling of Information..... 15–8
15.9. A Serial Half-adder..... 15–8
15.10. Serial Binary Adders..... 15–9
15.11. Serial Registers and Accumulators..... 15–10
15.12. Negative Numbers..... 15–11
15.13. A Serial Binary Accumulator. 15–12
15.14. Serial Binary Multiplication. 15–13
15.15. Processing of Negative Numbers..... 15–14
15.16. Serial Binary Division..... 15–15
15.17. Parallel Binary-arithmetic Units..... 15–17
15.18. Serial-parallel Arrangements 15–19
15.19. Decimal-arithmetic Operations..... 15–21
15.19.1. Addition and Subtraction 15–21
15.19.2. Multiplication and Division..... 15–21
15.20. Floating-point Numbers..... 15–22
15.20.1. Representation of Floating-point Numbers..... 15–22
15.20.2. Addition of Floating-point Numbers, General Discussion..... 15–22
15.20.3. Representation of Zero.. 15–23
15.20.4. Normalization..... 15–23
15.20.5. Floating-point Addition. 15–23
15.20.6. Floating-point Multiplication..... 15–24
15.20.7. Floating-point Speeds... 15–24
15.21. Structure of Arithmetic Units 15–24

A computer may do nearly 100,000 arithmetic operations per second, and a problem that may run for a week is not unusual. This is approximately 15×10^9 arithmetic operations. Each arithmetical operation probably involves at least 300 logical events. Therefore, in order to obtain correct answers something on the order of 10^{13} logical events must all happen correctly. This requires very careful design of the individual circuits. The general result of these requirements is that digital-computer circuits have been designed to operate in one of two states. That is, each individual circuit is either "on" or "off," except during a transition period.

This situation has naturally led to the design of digital computers operating in the binary number system. There are other advantages in the binary number system. For example, the arithmetic circuit is simpler and the memory space is used more efficiently.

Table 15.1. Binary, Octal, and Hexadecimal Representation of Integers

Decimal integer	Binary representation	Octal grouping	Octal representation	Hexadecimal grouping	Hexadecimal representation
0	00000000000	000 000 000 000	0000	0000 0000 0000	000
1	00000000001	000 000 000 001	0001	0000 0000 0001	001
5	00000000101	000 000 000 101	0005	0000 0000 0101	005
7	00000000111	000 000 000 111	0007	0000 0000 0111	007
8	00000001000	000 000 001 000	0010	0000 0000 1000	008
10	00000001010	000 000 001 010	0012	0000 0000 1010	00*u*
15	00000001111	000 000 001 111	0017	0000 0000 1111	00*z*
16	00000010000	000 000 010 000	0020	0000 0001 0000	010
35	00000100011	000 000 100 011	0043	0000 0010 0011	023
105	00001101001	000 001 101 001	0151	0000 0110 1001	069

On the other hand, the large number of characters needed to represent numbers of reasonable size makes it difficult for individuals to write and understand binary numbers. For this reason radices related to the binary system have been used. Most binary computers have a mode of input and output which uses the octal (radix 8) or the hexadecimal (radix 16) number systems. This grouping of binary bits is illustrated in Table 15.1.

15.1. THE BINARY NUMBER SYSTEM

The binary number system uses two digits or characters called "zero" and "one" and a positional notation analogous to the decimal number system. Successive

Table 15.2*a*. Binary Addition of Digits

	0	1
0	0	1
1	1	10

Table 15.2*b*. Binary Multiplication of Digits

	0	1
0	0	0
1	0	1

positions correspond to successive powers of 2. For example, 11010.101 means $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$. Table 15.2*a* gives an addition table for binary digits. The sum of two binary digits is at most "two," which is written as $10(1 \times 2^1 + 0 \times 2^0)$. Table 15.2*b* gives the product for pairs of binary digits. Contrast this with the decimal multiplication table one learns in grade school.

The simplicity of the binary number system is even more striking when one considers multiplication of binary numbers. In each position the multiplicand is either added or not added. In division at each position the divisor divides the numerator either once or not at all. These features make the electronic circuitry much simpler than for decimal-arithmetic circuits.

15.2. THE OCTAL NUMBER SYSTEM

The binary bits of a number may be considered in groups of three, and other symbols may be used to represent the eight different combinations that can occur. Usually, the groups are interpreted as binary integers and the corresponding decimal digits are used. Thus, the binary number 101100.011 would be written as 54.3, which stands for the decimal number 44⅜. The digit-by-digit addition and multiplication tables for the octal number system are given in Table 15.3.

Table 15.3*a*. Octal Addition

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Table 15.3*b*. Octal Multiplication

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

The advantage of the octal system compared with the binary system is that it gives a more compact representation of numbers. The octal grouping of the binary bits usually occurs only during input and output; so there is no added complication of arithmetic circuits.

15.3. THE HEXADECIMAL NUMBER SYSTEM

If the bits of a binary number are considered in groups of four a compact notation is obtained. In this case the base is 16 and 16 symbols are required to represent the

Table 15.4. Hexadecimal Multiplication Table

	0	1	2	3	4	5	6	7	8	9	*u*	*v*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	*u*	*v*	*w*	*x*	*y*	*z*
2	0	2	4	6	8	*u*	*w*	*y*	10	12	14	16	18	1*u*	1*w*	1*y*
3	0	3	6	9	*w*	*z*	12	15	18	1*v*	1*y*	21	24	27	2*u*	2*x*
4	0	4	8	*w*	10	14	18	1*w*	20	24	28	2*w*	30	34	38	3*w*
5	0	5	*u*	*z*	14	19	1*y*	23	28	2*x*	32	37	3*w*	41	46	4*v*
6	0	6	*w*	12	18	1*y*	24	2*u*	30	36	3*w*	42	48	4*y*	54	5*u*
7	0	7	*y*	15	1*w*	23	2*u*	31	38	3*z*	46	4*x*	54	5*v*	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1*v*	24	2*x*	36	3*z*	48	51	5*u*	63	6*w*	75	7*y*	87
u	0	*u*	14	1*y*	28	32	3*w*	46	50	5*u*	64	6*y*	78	82	8*w*	96
v	0	*v*	16	21	2*w*	37	42	4*x*	58	63	6*y*	79	84	8*z*	9*u*	*u*5
w	0	*w*	18	21	30	3*w*	48	51	60	6*w*	78	81	90	9*w*	*u*8	*v*4
x	0	*x*	1*u*	27	34	41	4*y*	5*v*	68	75	82	8*z*	9*w*	*u*9	*v*6	*w*3
y	0	*y*	1*w*	2*u*	38	46	54	62	70	7*y*	8*w*	9*u*	*u*8	*v*6	*w*4	*x*2
z	0	*z*	1*y*	2*x*	3*w*	4*v*	5*u*	60	78	87	96	*u*5	*v*4	*w*3	*x*2	*y*1

digits in any position. The digits zero through nine are used in their usual sense and the other six digits have been represented by the letters u, v, w, x, y, and z or by a, b, c, d, e, and f. Both the addition table and the multiplication table (see Table 15.4) have become more formidable now. The arithmetic may all still be done in the binary number system, and on input and output translation can be made automatic, so perhaps only the service engineer will ever have occasion to do arithmetic in this number system.

The hexadecimal system requires six extra characters. On the other hand, the input and output equipment must usually handle decimal digits. It takes at least four binary digits to represent the decimal digits 0, 1, . . . , 9 (three, as shown above, can represent only the digits 0, 1, 2, . . . , 7). Thus, if decimal input and output is required it takes relatively little more to have hexadecimal input and output.

15.4. TERNARY NUMBER SYSTEMS

Ternary (base three) number systems have been proposed but have not been used very extensively. A base three computer called SETUN was put into operation at the University of Moscow, U.S.S.R., in 1959. This computer uses two binary elements to store each ternary digit.

15.5. REPRESENTATION OF DECIMAL DIGITS

In order to attain maximum reliability and consequent maximum error-free computation runs most digital computers have been designed out of component circuits with only two states. This has required that decimal digits be represented by a

Table 15.5. Binary-decimal Coding

Binary code	Binary-decimal code	Excess-three code	1242′ code	Biquinary for decimal digits
0000	0	. . .	0	10000 10
0001	1	. . .	1	01000 10
0010	2	. . .	2	00100 10
0011	3	0	3	00010 10
0100	4	1	4	00001 10
0101	5	2	. . .	10000 01
0110	6	3	. . .	01000 01
0111	7	4	. . .	00100 01
1000	8	5	. . .	00010 01
1001	9	6	. . .	00001 01
1010	. . .	7	. . .	
1011	. . .	8	5	
1100	. . .	9	6	
1101	. . .	. . .	7	
1110	. . .	. . .	8	
1111	. . .	. . .	9	

combination of binary digits or bits. In order to give 10 different digits it is necessary to use at least 4 bits. However, 4 bits can represent 16 different symbols, and this usually results in an inefficient use of the memory in the computer. In some systems the six unused codes may serve some purpose, for example, detection of errors; so this may not represent a total loss.

In some computers more than 4 bits have been used to represent the decimal digits. The extra redundancy makes possible checking, and even error correction.

The number of ways that the sixteen 4-bit codes can be assigned to represent the ten decimal digits is on the order of 10^{10} ($_{16}P_{10}$). Obviously very few of these ways have been investigated for their merits. The characteristics that affect a decision are

(1) simplicity of the arithmetic circuits, (2) the ease of handling negative numbers (complementation), and (3) decoding and encoding in the terminal equipment.

The coding systems in most extensive use are the (1) simple binary, (2) the excess-three binary, and the 1242′ system. These systems are all shown in Table 15.5. Note that in these three systems the codes are assigned sequentially with respect to the binary numbers but the gap of six occurs at different positions.

Another widely used system for binary representation of decimal digits is the biquinary system. Here 7 bits are used to represent each decimal digit. The 7 bits are divided into two groups of two and five; the bits in the group of two represent 0 or 5, and the bits in the group of five represent the digits 0, 1, 2, 3, and 4. Thus each decimal digit is represented by exactly 1 bit being present in each group. See Table 15.5.

15.6. NOTATION AND SYMBOLS

Before embarking upon the description of binary-arithmetic circuits a uniform notation and set of symbols will be described. The following discussion will be in terms of static flip-flops. An analogous description in terms of dynamic flip-flops could easily be done.

15.6.1. Flip-flop Notation. Figure 15.1 shows a static flip-flop with two outputs and two inputs. If the flip-flop is called A then the two outputs are called A and nA (read "A" and "not A"). The two inputs are the set input and the reset input. The flip-flop is in the "true" state or "1" state if the output A is at a high potential and nA is at a low potential. The flip-flop is said to be in the "false" state or "0" state if nA is high and A is low.

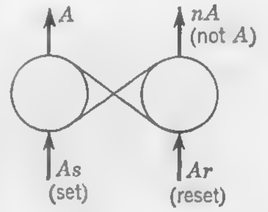


FIG. 15.1. Flip-flop.

After a signal has been received on the set input, A is high and nA is low; that is, the flip-flop is in the 1 state. Likewise, a signal on the reset input puts the flip-flop into the 0 state.

As mentioned in the section on flip-flops, there can be a third input to the flip-flop which will cause it to change its state. Such inputs are more critical in operation; so they will not be used in the circuits discussed here.

15.6.2. AND-gate Notation. Figure 15.2 shows a three-input AND gate. If the inputs are labeled A, B, and C, and the output D, then the operation of the circuit can be expressed by the Boolean equation $D = A \cdot B \cdot C$. That is, if A, B, and C are high (=1), then D is high (=1). On the other hand, if one or more of the inputs is low (=0), then $D = 0$.

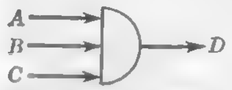


FIG. 15.2. AND gate.

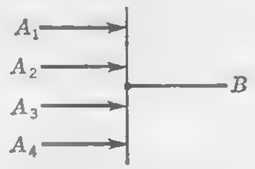


FIG. 15.3. OR gate.

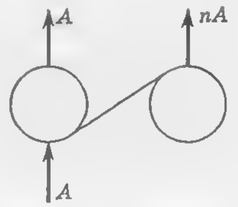


FIG. 15.4. Inverter.

In the logical circuitry discussed here no attention will be paid to loading or frequency response; so AND gates with any number of inputs will be allowed.

15.6.3. OR-gate Notation. Figure 15.3 shows a four-input OR gate. If the inputs are called A_1, A_2, A_3, and A_4 and the output is B, then the effect of the circuit is given by the Boolean equation $B = A_1 + A_2 + A_3 + A_4$. That is, if any input is high (=1), then the output is high (=1).

From the logical point of view there is no limit to the number of inputs for each logical circuit and there is no limit to the length of chains of AND gates and OR gates. In practice, loading considerations and frequency response limit this.

15.6.4. Inverters. In many situations it is convenient to derive the false signal from the true signal or vice versa. Figure 15.4 shows an inverter which is similar to

the flip-flop in that it has true and false outputs but only one input. If the inverter is labeled A and there is a signal on the input (the input is high or equal to 1), then A is high and nA is low (equal to 1 and 0, respectively). If the input is low, then A is low (false, or equal to 0), and nA is high.

In practice, inverters may also be used to restore proper d-c levels in the circuitry. Frequently a schmidt-trigger circuit is used.

15.6.5. Pulse Designations. *Clock Pulse.* In this section the clock pulse is the basic timing pulse of the system. It will be assumed to be a continuous equally

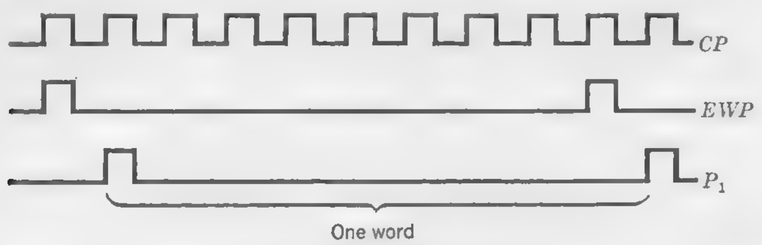


FIG. 15.5. Pulse designations.

spaced pulse train. Generally it will be the highest pulse frequency occurring in the system. In some cases it will be necessary to consider several phases of clock pulses; these will be referred to as CP_1, CP_2, etc.

In serial systems the clock pulses CP will be divided into equal groups and referred to as *words*. The kth pulse in any word group will be referred to as P_k. Thus the first pulse is P_1 and the last pulse will be called EWP (end-of-word pulse). Figure 15.5 shows the relative timing of pulses in a serial system.

15.6.6. Delay Lines. In high-frequency pulse systems compensating delays may be required to make circuits operate reliably. In this section such considerations will be ignored and only those delays that are logically necessary will be considered. Figure 15.6 shows a delay-line symbol. The letter alongside the delay line denotes the amount of delay in terms of clock pulses. Fractional values of delay are permissible. If $P(t)$ denotes the pulse train occurring at the input A at time t, then $P(t-k)$ will appear at the output B of a delay line of length k clock pulses.

FIG. 15.6. Delay line.

15.6.7. Timing Considerations. In order to ignore frequency response of circuits and compensating delays mentioned above, the response of chains of AND gates, OR gates, and inverters will be assumed to be instantaneous. If a flip-flop changes state then its outputs are effective in their new configuration not earlier than one clock-pulse interval later. That is, there is a deliberate delay between the input and output of flip-flops so that transitions occurring at one clock-pulse time will not have effect until the following clock pulse.

15.7. COUNTERS

Figure 15.7 shows a four-stage static binary counter. Note that all gates are qualified by the clock pulse CP; so no flip-flop will change its state, except at the clock time. The CP is also the input to the counter, so that it goes through its various configurations every 16 clock times.

The counter is completely described from the logical point of view by either the diagram or the equations (Fig. 15.7).

The techniques of logical design will be utilized to modify the above counter to become a decimal counter. That is, the counter must reset after 10 counts instead of after 16.

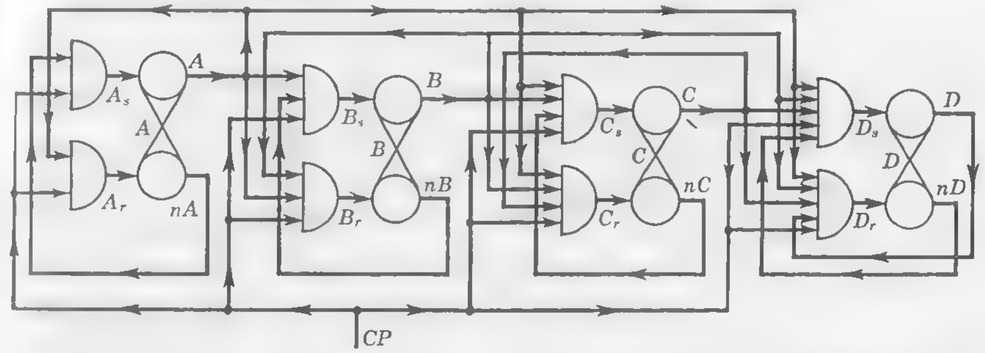
15.7.1. A Decimal Counter. Table 15.6 shows the set and reset signals for both the binary counter and for a decimal counter. The 0's mean no signal, 1's mean a pulse, and the X means that it does not matter. This last comes from the fact that if a

flip-flop is in the reset state another reset signal causes no change. This is true in actual practice since any transients introduced by trying to reset a flip-flop already in the reset state will die out by the next clock-pulse time.

Table 15.6. The Set and Reset Signals for a Four-stage Binary and a Four-stage Decimal Counter

ABCD	Binary counter								Decimal counter								Simplified			
	As	*Ar*	*Bs*	*Br*	*Cs*	*Cr*	*Ds*	*Dr*	*As*	*Ar*	*Bs*	*Br*	*Cs*	*Cr*	*Ds*	*Dr*	*Bs*	*Br*	*Ds*	*Dr*
0000	1	0	0	*X*	0	*X*	0	*X*	1	0	0	*X*	0	*X*	0	*X*	0	0	0	0
1000	0	1	1	0	0	*X*	0	*X*	0	1	1	0	0	*X*	0	*X*	1	0	0	1
0100	1	0	*X*	0	0	*X*	0	*X*	1	0	*X*	0	0	*X*	0	*X*	0	0	0	0
1100	0	1	0	1	1	0	0	*X*	0	1	0	1	1	0	0	*X*	0	1	0	1
0010	1	0	0	*X*	*X*	0	0	*X*	1	0	0	*X*	*X*	0	0	*X*	0	0	0	0
1010	0	1	1	0	*X*	0	0	*X*	0	1	1	0	*X*	0	0	*X*	1	0	0	0
0110	1	0	*X*	0	*X*	0	0	*X*	1	0	*X*	0	*X*	0	0	*X*	0	0	0	0
1110	0	1	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	1	1	0
0001	1	0	0	*X*	0	*X*	*X*	0	1	0	0	*X*	0	*X*	*X*	0	0	0	0	0
1001	0	1	1	0	0	*X*	*X*	0	0	1	0	*X*	0	*X*	0	1	0	0	0	1
0101	1	0	*X*	0	0	*X*	*X*	0	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	0	0	0	0
1101	0	1	0	1	1	0	*X*	0	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	0	1	0	1
0011	1	0	0	*X*	*X*	0	*X*	0	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	0	0	0	0
1011	0	1	1	0	*X*	0	*X*	0	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	0	0	0	0
0111	1	0	*X*	0	*X*	0	*X*	0	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	0	0	0	0
1111	0	1	0	1	0	1	0	1	*X*	*X*	*X*	*X*	*X*	*X*	*X*	*X*	0	1	1	0

Referring to Table 15.6, observe that the set and reset signals are the same in the two cases for counts 0 through 8. Since states 10 through 15 cannot occur in the decimal counter all these positions are marked with X's. The signals for flip-flops A and C are the same, even for count 9. Therefore, to make a decimal counter only the



$A_s = nA \cdot CP$ $\quad$ $B_s = A \cdot nB \cdot CP$ $\quad$ $C_s = A \cdot B \cdot nC \cdot CP$ $\quad$ $D_s = A \cdot B \cdot C \cdot nD \cdot CP$

$A_r = A \cdot CP$ $\quad$ $B_r = A \cdot B \cdot CP$ $\quad$ $C_r = A \cdot B \cdot C \cdot CP$ $\quad$ $D_r = A \cdot B \cdot C \cdot D \cdot CP$

FIG. 15.7. A four-stage binary counter.

logic for the inputs to flip-flops B and D must be changed. The complete equations for the inputs to B and D are (1) $Bs = A \cdot nb \cdot nC \cdot nD + A \cdot nB \cdot C \cdot nD$, (2) $Br = A \cdot B \cdot nC \cdot nD + A \cdot B \cdot C \cdot nD$, (3) $Ds = A \cdot B \cdot C \cdot nD$, and (4) $Dr = A \cdot nB \cdot nC \cdot D$. By replacing some of the X's by 1's these equations can be simplified to (*a*) $Bs = A \cdot nB \cdot nD$, (*b*) $Br = A \cdot B$, (*c*) $Ds = A \cdot B \cdot C$, (*d*) $Dr = A \cdot nC$. To simplify the presentation the clock pulses have been omitted from all the above equations.

15.8. SERIAL AND PARALLEL HANDLING OF INFORMATION

The simpler digital computers handle information in the form of pulse trains on single leads. In this type of system the minimum time to process a number is the number of clock-pulse times that correspond to the length of the number. In order to attain higher speed parallel channels have been used up to the point where there is a separate channel for each binary position in the numbers. Such a parallel system is substantially faster but also involves much more equipment. For example, a parallel system may have forty binary adders in the arithmetic circuits whereas a serial machine will have only one adder.

This section considers serial circuits first, parallel circuits second, and serial-parallel combinations third. The static counter of Sec. 15.7 counts pulses arriving on a single channel but has a flip-flop for each binary position in the number; so it is essentially a parallel device. In the next section a binary counter will be considered which is not parallel.

15.9. A SERIAL HALF-ADDER

Figure 15.8 shows the diagram of a serial half-adder. Given any binary number (of any length) arriving on lead N the output S is a number one unit larger provided there was a signal on the count input during the first pulse position of the number. This is called a half-adder, since two such circuits can be used as a complete binary adder.

If counting is to occur the flip-flop C must be set during the first pulse period of the number N. Simple analysis will show that if the first bit of N (least significant bit) is

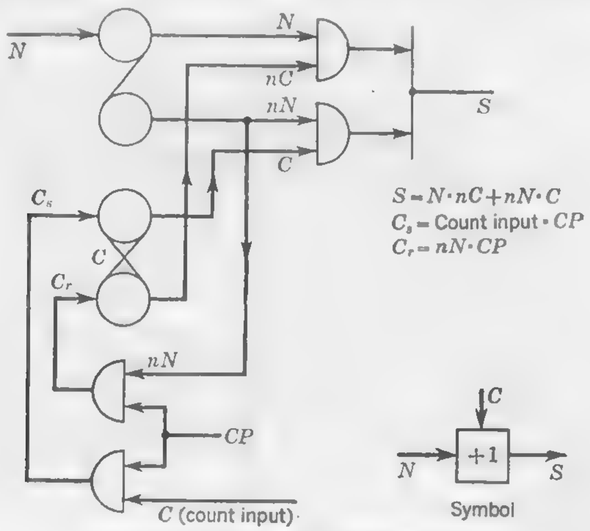


FIG. 15.8. Serial half-adder.

a 1 the sum digit should be a 0, and the carry flip-flop C should remain set. In fact the operation of the counter can be specified by the following truth table:

N	C	S	Cr
0	0	0	X
0	1	1	1
1	0	1	X
1	1	0	0

or by the equations

$$S = N \cdot nC + nN \cdot C$$
$$Cr = nN$$

In subsequent sections it may be necessary to have half-adders in the circuitry. In such cases they may be represented by a simpler symbol as shown in the lower right-hand corner of Fig. 15.8.

15.10. SERIAL BINARY ADDERS

Another information input can be combined with the half-adder circuit of Fig. 15.8 to produce the binary adder of Fig. 15.9. The truth table is as follows:

A	B	C	S	Cs	Cr
0	0	0	0	0	X
1	0	0	1	0	X
0	1	0	1	0	X
0	0	1	1	0	1
1	1	0	0	1	0
1	0	1	0	X	0
0	1	1	0	X	0
1	1	1	1	X	0

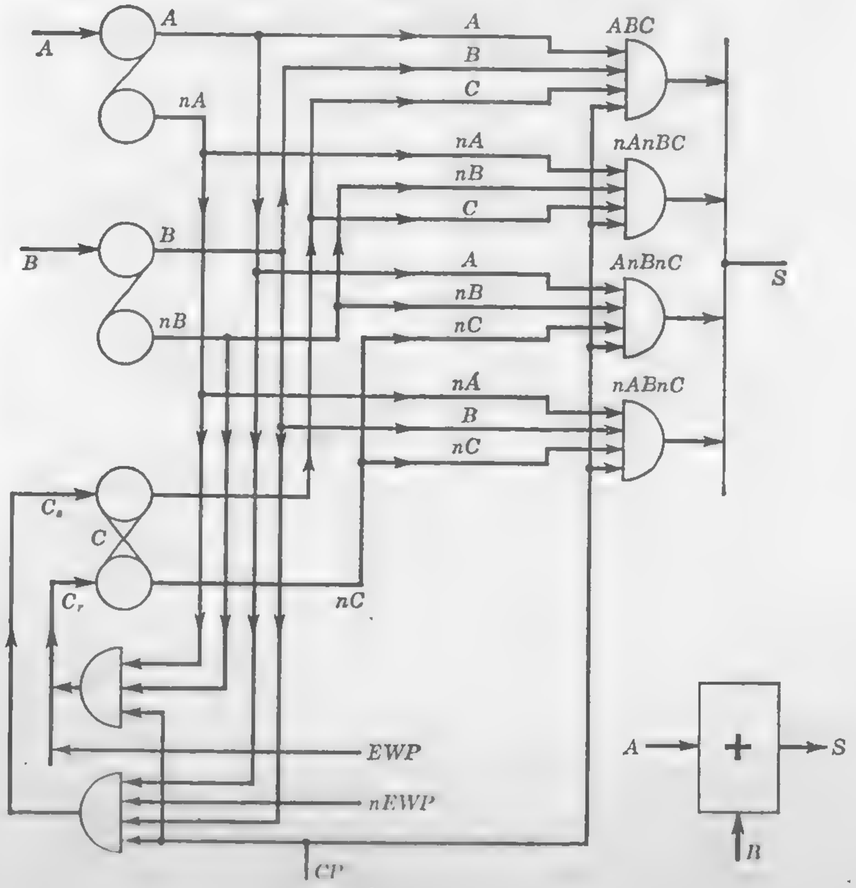


FIG. 15.9. Serial binary adder.

The concept of carry suppression is introduced here, although it will not be needed until negative numbers are considered. Carry suppression is obtained by resetting the carry flip-flop at the end of the number (EWP time). The equations specifying the operation of the adder are

$$S = nA \cdot nB \cdot C + A \cdot nB \cdot nC + nA \cdot B \cdot nC + A \cdot B \cdot C$$
$$Cs = A \cdot B \cdot nEWP \cdot CP$$
$$Cr = nA \cdot nB \cdot CP + EWP$$

An alternate approach to a serial binary adder is shown in Fig. 15.10. Here, for a given clock frequency, the circuits must respond substantially faster but there is less

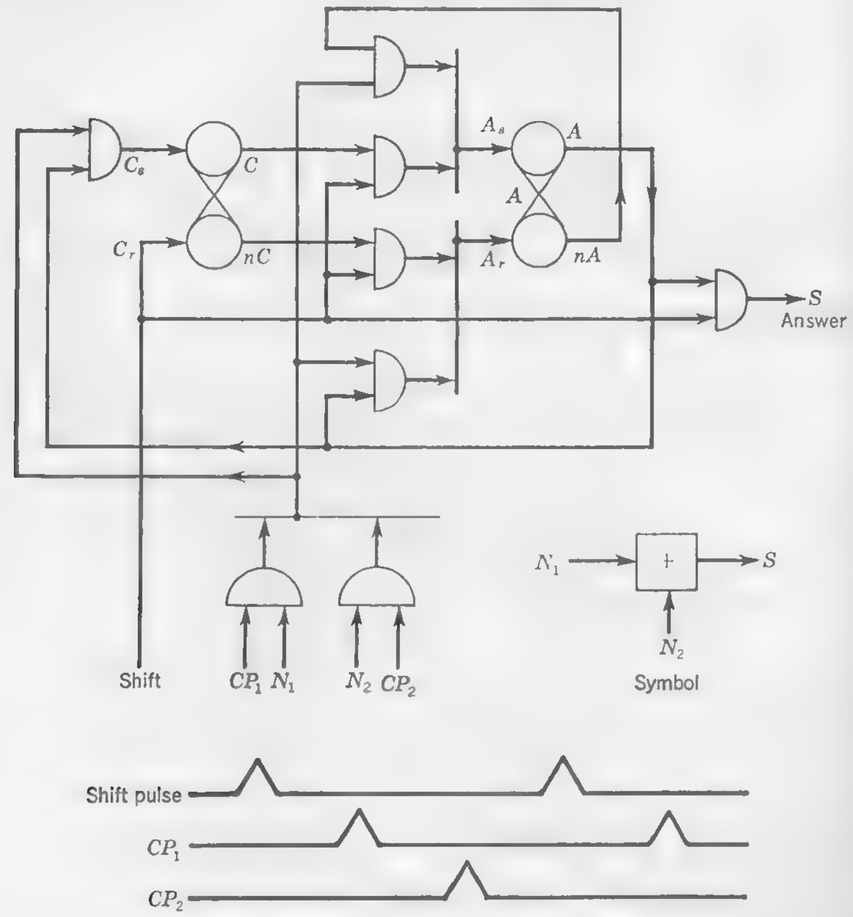


FIG. 15.10. Shifting serial binary adder.

total equipment involved. Starting with the shift pulse the carry digit is transferred from flip-flop C to flip-flop A, and the answer is transferred from A to S. Now the circuit is a two-stage binary counter with the incoming carry already in A. Digits N_1 and N_2 arrive over the same channel at different times and produce a maximum count of 3 in the two flip-flops. A shift pulse brings out the answer digit and the cycle starts for the next more significant position.

15.11. SERIAL REGISTERS AND ACCUMULATORS

Information may be stored in a delay line by having amplifiers and pulse-reshaping circuits in the circulation loop. A logical circuit showing input, clear, and output

terminals is shown in Fig. 15.11*b*. The delay line may be of any length. As part of the main memory of the computer it may store several words (even more than 100 words in some cases), or it may store only one word or number. Such a memory unit which stores one or two words or numbers will be called a register.

A binary adder may be included in the recirculation loop, and the register becomes an accumulator (see Fig. 15.11*a*). Since the length of the delay line is immaterial several different numbers may be added in the same accumulator without interference,

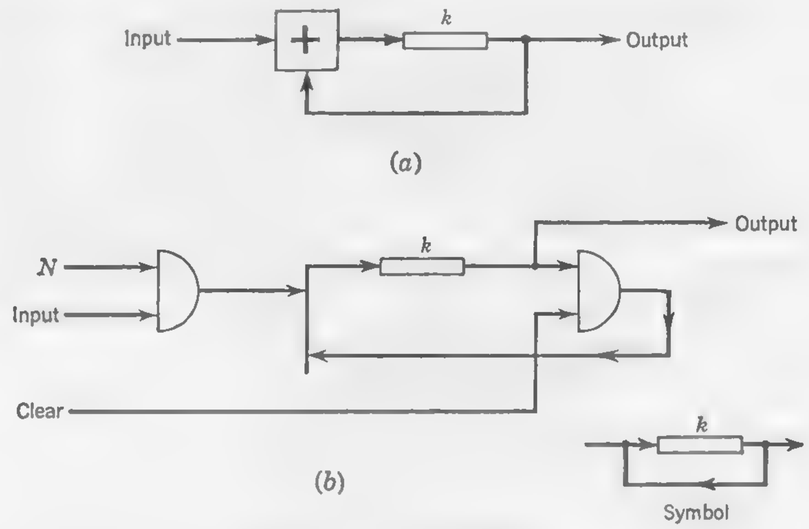


FIG. 15.11*a*. An accumulator.
FIG. 15.11*b*. Register.

provided that the numbers are positive. The next section considers negative numbers. Then an accumulator which will handle positive and negative numbers will be presented.

15.12. NEGATIVE NUMBERS

In most computers negative numbers are handled in complementary form. In decimal computers the complements may be with respect to either 10^k or to $10^k - 1$ where there are k digits in the numbers. These two complements are referred to as the 10's and 9's complements, respectively. Likewise, in the binary number system, complements with respect to 2^k or with respect to $2^k - 1$ are used. Similar to the decimal terminology these are called 2's and 1's complements.

If negative numbers are handled as either 10's or 2's (depending on the number base) the positive and negative numbers may be added using the normal rules for addition. Furthermore, ordinary adders such as those described in Sec. 15.10 will add positive and negative numbers, provided that carry is suppressed at the end of the word (number).

1's or 9's complements may be used to represent negative numbers, and the above statements are true provided the carry out of the most significant position is added into the least significant position. This process is called "end-around carry." Without carry suppression this end-around carry can be made to happen automatically in an accumulator such as that shown in Fig. 15.11*a*.

Figure 15.12 shows a binary complementing circuit. If the number arriving at the complementer is called N the signal nN is the 1's complement. To obtain the 2's complement unity must be added to this number. Another method of obtaining the result is to start the 1's complementing of N after the first nonzero digit. The circuit of Fig. 15.12 is based upon this rule.

15.13. A SERIAL BINARY ACCUMULATOR

Making use of the circuits described above, an accumulator may be constructed which is capable of doing addition and subtraction of signed numbers. Such a circuit is shown in Fig. 15.13. There is an input for the numbers N and output S and four "command" lines which control the activity of the accumulator.

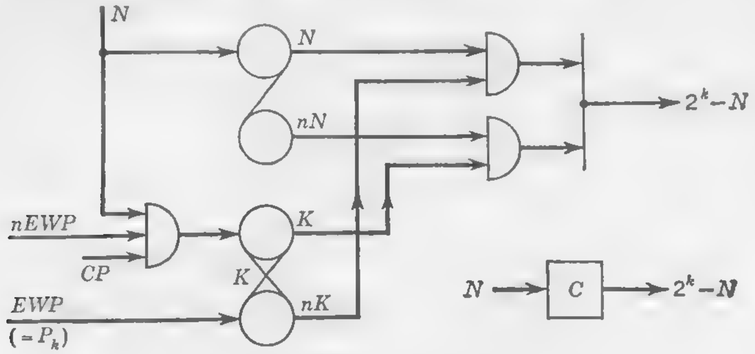
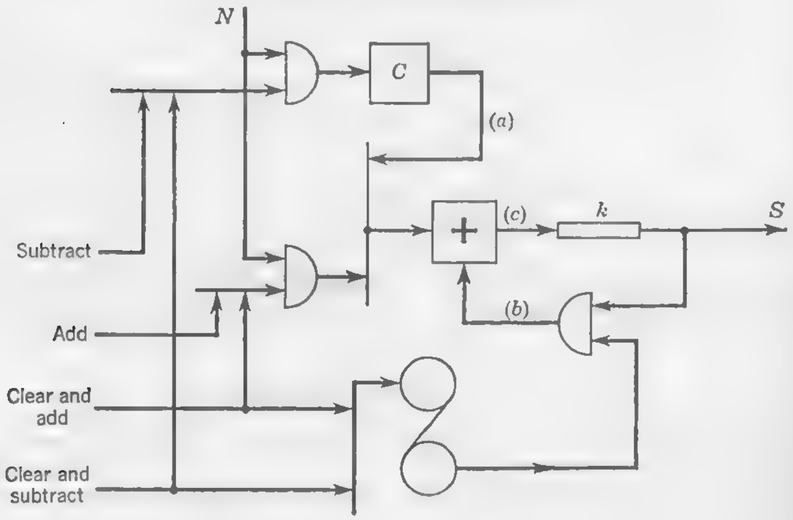


FIG. 15.12. Two's complementer.



Numerical example

Point in circuit	Commands			
	Clear and add	Clear and subtract	Add	Subtract
N	01010	01010	01010	01010
(a)	00000	10110	00000′	10110
(b)	00000	00000	01011	01011
(c)	01010	10110	10101	00001

FIG. 15.13. Serial binary accumulator.

The SUBTRACT and the CLEAR-AND-SUBTRACT signals open the gate which allows the incoming information to flow through the complementer. The ADD and the CLEAR-AND-ADD signals open the gate which allows N to pass directly to the adder. Both clear signals pass through the inverter and operate the clear gate, which is in the recirculation loop of the accumulator. The delay line stores k digits, and it is assumed that there is carry suppression (or end-around carry) in the adder and reset

ting of the complementer occurs at the proper times. That is to say, it is assumed that EWP comes every kth clock pulse and that any numbers N to be added or subtracted arrive in the right phase.

An actual example with $k = 5$ is given on the bottom of Fig. 15.13 for the four possible commands. The binary bits listed are those that pass the corresponding point in the circuit during one word time. The number in the accumulator register can be interpreted as the integer 11 and the number N as the integer 10. Likewise, they could be interpreted as the fractions $^{11}/_{32}$ and $^{10}/_{32}$.

The point N can be visualized as the terminal labeled "output" in the circuits of Fig. 15.11*a* or *b*.

15.14. SERIAL BINARY MULTIPLICATION

The binary multiplication table is much simpler than the decimal table (cf. Table 15.2*b*). Since the digits are either 0 or 1, the product of two digits is 0, unless both are 1, in which case the product is 1. The simplest multiplication circuits are those which multiply by repeated addition. Consider the following example showing the multiplication of 5 (0101) by 6 (0110):

```
  0110
  0101
 ------
  0110
 0000
0110
0000
--------
00011110
```

In this example four words (numbers of fixed length) are involved, the multiplicand (0110), the multiplier (0101), and two words to store the product (0001 1110). In order to conserve storage space the multiplier will be stored in part of the product register and its digits will be deleted as they are used in order to make room for the product. Also, the order of processing the digits of the multiplier will be reversed; that is, the most significant digits will be used first. The following example is a rearrangement of the above problem to illustrate these details:

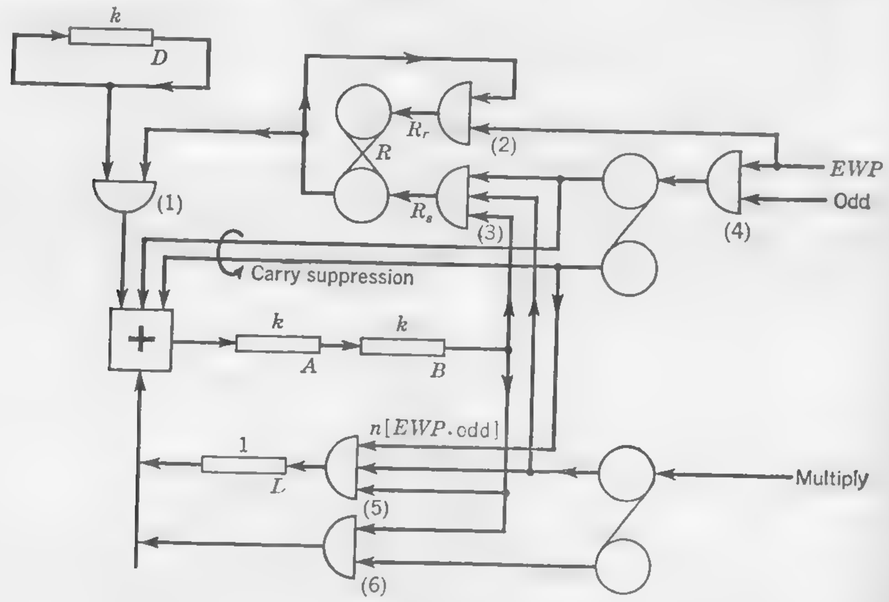
```
0110
0000      0 1 0 1
-----    ---------
 000     0|1 0 1
  01     1 0|0 1
   0     0 0 0|1
               0 1 1 0|
-----    ---------
0001     1 1 1 0
```

The zigzag line marks the division between the partial products and the digits of the multiplier. The four contributions of the partial products are displayed separately in the example, whereas in the circuit of Fig. 15.14 they are accumulated as they are developed.

The multiplier of Fig. 15.14 works only for positive numbers as it stands. Multiplication of positive and negative numbers will be considered in the next section.

The example shown on the bottom of Fig. 15.14 shows what happens from one word time before multiplication starts until one word time after it terminates. The digits of the multiplier in the order of most significant first are picked up and stored in flip-flop R, where they control whether or not the multiplicand in register D is to be added to the partial products. The accumulator is two words long in order to store the full double-word-length product. During the multiplication process the partial products must be shifted relative to the multiplicand in order to account for

the various weights of the multiplier digits. This shifting is accomplished by inserting a one-clock-pulse delay L in the recirculation loop. The digits of the multiplier are inspected at gate 3 (Rs), and they are simultaneously deleted at gate 5 before the delay line L. This inspection and deletion must occur every other word time at the end of the word. The signal called "odd" comes from one side of a flip-flop continuously driven by, say, EWP. Thus it changes its state each word time and consequently marks odd and even word times.



Example

Word time	Status at end of word time			Status during word time	
	L	A	B	R	Multiply
−1	0	0101	0000	0	No
0	0	0000	0101	0	Yes
1	0	1010	0000	0	Yes
2	0	0000	1010	0*	Yes
3	0	0100	0000	0	Yes
4	0	0110	0100	1*	Yes
5	0	1000	0110	0	Yes
6	0	1100	1000	0*	Yes
7	0	0000	1100	0	Yes
8	1	1110	0000	1*	Yes
9	0	0001	1110	0	Yes
10	0	1110	0001	0	No

FIG. 15.14. Serial binary multiplier.

In Fig. 15.14 the digits of the multiplier are underlined in columns A and B, and in the column labeled R the multiplier digits are starred as they appear there. Other arrangements of the adder, the storage registers, and the units delay are possible. In fact, a somewhat different arrangement is shown in Fig. 15.15.

In the binary adder of Fig. 15.9 carry suppression is required each word time. In the two-word accumulator required in the multiplier carry suppression must occur every other word time.

15.15. PROCESSING OF NEGATIVE NUMBERS

Most computers make use of complements in order to accomplish the addition and subtraction of positive and negative numbers. The numbers may be stored in the

memory either in sign-and-absolute-value or in complementary form. That is, positive numbers have a 0 in the sign position and the digits occurring in normal fashion. By sign-and-absolute-value form it is meant that negative numbers differ from positive numbers only in the sign position. In complementary form the negative numbers have a 1 in the sign position and the rest of the number is in the form $2^k - N$ or $2^k - 1 - N$. When numbers are stored in complementary form it is convenient to let the digit at the most significant end represent the sign. In the sign-and-absolute-value convention the sign may be on either end of the number.

Note that it is still possible to let the sign be on the least significant end of the number, when processing numbers in complementary form, by doing the carry suppression at the right time.

Multiplication and division are most easily done on numbers in sign-and-absolute-value form. If they are in complementary form and some of the operands are negative, then corrections must be added to products. This is illustrated by the following formulas:

Factors	Result produced by binary multiplier (Sec. 15.14)	Desired result
a, b	ab	ab
$a, 2^k - b$	$a2^k - ab$	$2^{2k} - ab$
$2^k - a, b$	$b2^k - ab$	$2^{2k} - ab$
$2^k - a, 2^k - b$	$2^{2k} - a2^k - b2^k + ab$	ab

Thus, after the multiplication of the operands, one or both of the factors may have to be subtracted, or added, to the most significant half of the product. Computers have been built that multiply complements, but this difficulty has caused many of them to be designed to multiply absolute values, and extra circuitry has been used to take care of the sign.

15.16. SERIAL BINARY DIVISION

Division by trial subtraction, which is the method taught in grade schools, is easy to mechanize in the binary number system. In each position the divisor either goes into the numerator or it does not go. If overdraft is allowed, two k binary digit numbers can be divided in k word times.

Division occurs much less frequently in most problems than do additions, subtractions, and multiplications. Figures on the order of 5 per cent have been mentioned as the relative frequency of divisions. Thus it is usually not important to mechanize a particularly fast division. Being consistent with this last remark, the multiplier of Fig. 15.14 is modified to accomplish both multiplication and division (see Fig. 15.15).

The elements of the two-word accumulator have been rearranged, and the carry-suppression circuits have been changed, since in division, as shown here, carry must be suppressed each word time. The circuitry that must be added to accomplish division is shown with dashed lines. The first six AND gates carry the same number designation as they do in Fig. 15.14.

The lines labeled MULT and DIV are the control lines which are high during the multiplication or during the division process. P_1 (cf. Fig. 15.5) is a pulse coincident with the first pulse during the word time; that is, it occurs just after EWP.

In order to work properly, and this is true of most division circuits, the denominator must be greater than the numerator. Table 15.7 shows the contents of the numerator register N and the quotient register Q during the division of 5 by 6. Only 4 bits are used in the illustration. In this system the last event happening to the quotient

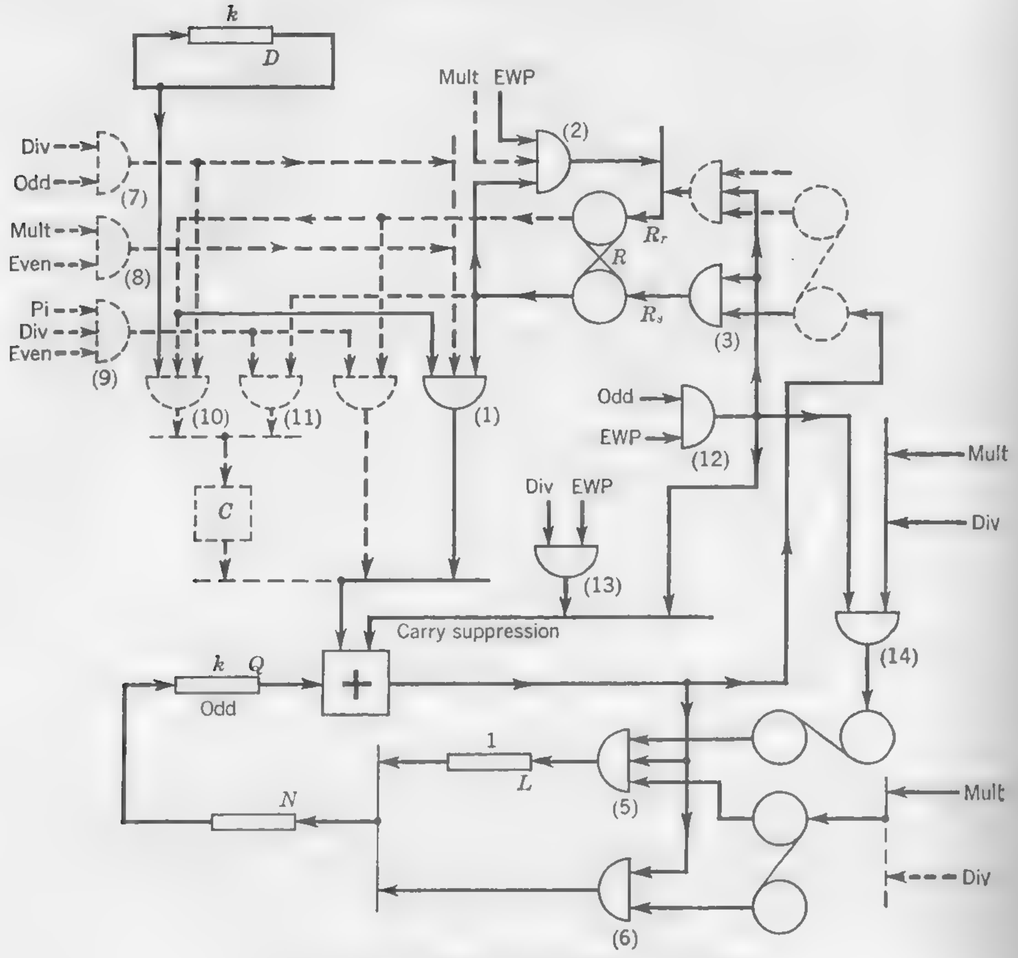


FIG. 15.15. Serial binary multiplier and divider.

Table 15.7. Division Example

Word time	N	Q	L	R	
−1	0000	0101			← Division starts here
0	0101	0000	0	0	
1	0010	0101	0	0	
2	1110	0010	0	1	
3	0010	1110	0	1	
4	1000	0010	0	0	
5	0110	1000	0	0	
6	0100	0110	0	0	
7	1110	0100	0	0	← Division ends here
8	0100	1110			
		Quotient			

is a shift; thus the last digit is always a 0. Depending upon overdraft or not, the next following operation is to add or to subtract 1 in this last position. Therefore, the last digit being 0 need not be considered as undesirable. The quotient produced in Table 15.7 is (1110), which interpreted as a fraction is $\frac{1}{2} + \frac{1}{4} + \frac{1}{8}$ or $\frac{7}{8}$. The answer is $\frac{5}{6}$, which lies between $\frac{13}{16}$ and $\frac{7}{8}$ (1101 and 1110).

15.17. PARALLEL BINARY-ARITHMETIC UNITS

In the parallel accumulator all the digits are available for processing simultaneously. Figure 15.16 indicates the general arrangement of the circuitry for a four-channel illustrative accumulator. The M register is a buffer register between the memory of the computer and the arithmetic unit; the A register stores the result of the last operation and will also store the result of the present operation. Except for the memory buffer register this accumulator is functionally similar to that of Fig. 15.13.

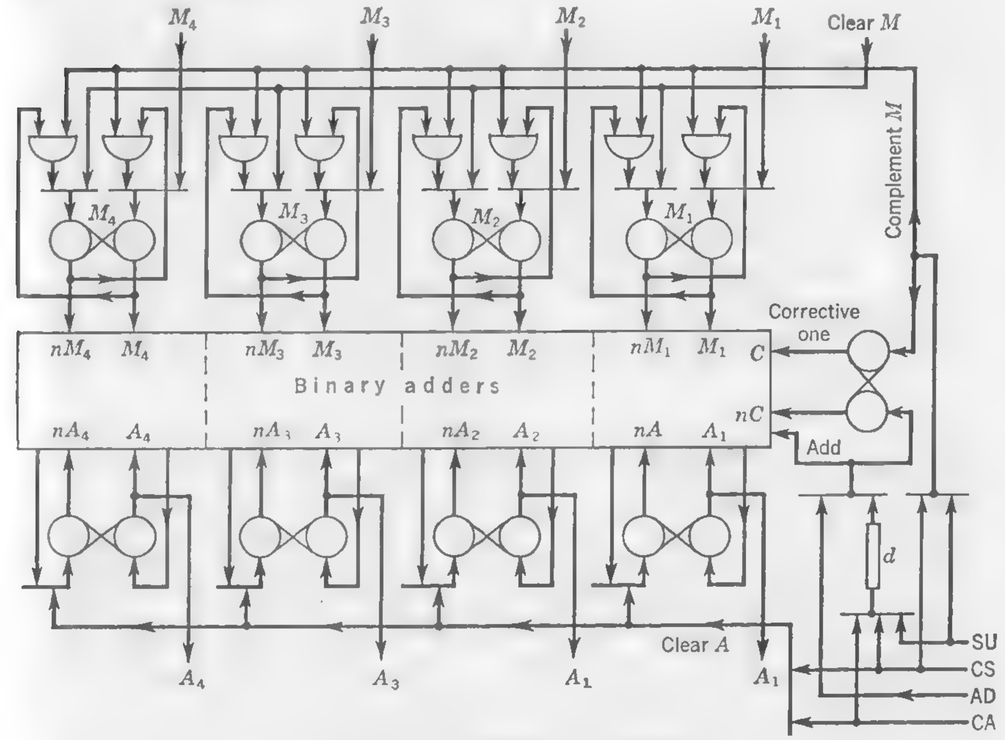


FIG. 15.16. Parallel binary accumulator.

The four commands [subtract (SU), clear and subtract (CS), add (AD), and clear and add (CA)] are shown in the lower right-hand corner of Fig. 15.16. The memory buffer register M is along the top of the diagram and the accumulator register A is along the bottom. The four information inputs to the M register are at the top and are labeled M_1, M_2, M_3, and M_4. Similarly, the outputs of the A register are labeled A_1 to A_4.

Both clear commands operate the clear A line. The subtract commands operate the complement M line which produces a 1's complement. To obtain a 2's complement unity must be added in the least significant position. This is accomplished by inserting a signal in the carry input to the addition circuits of the first stage. This is called a "corrective-1" (cf. Sec. 15.12).

Operational Sequence. First the M register is cleared; then the new operand is transferred into the M register. After the carry circuits for the whole accumulator have reached their final configuration (any transients have died out) the addition pulse can be applied, which establishes the result in the A register. If the command is SU, CS, or CA then time for a new carry propagation must be allowed between either the complementation or the clearing. This time is controlled by the delay d in Fig. 15.16.

The accumulator of Fig. 15.16 works for numbers stored in the memory with negative numbers in complementary form. If numbers are in sign-and-absolute-value

form then more logic is required with respect to the complement M line in order to process negative numbers correctly.

The speed of a parallel accumulator depends upon how long it takes the carry signal to propagate from one end to the other. In a vacuum-tube system with two tubes between each adder (because of the signal inversion) propagation times of less than 4 μsec have been accomplished for 37 binary bits. In such a system time must be allowed between each addition in the repeated additions of a multiplication to

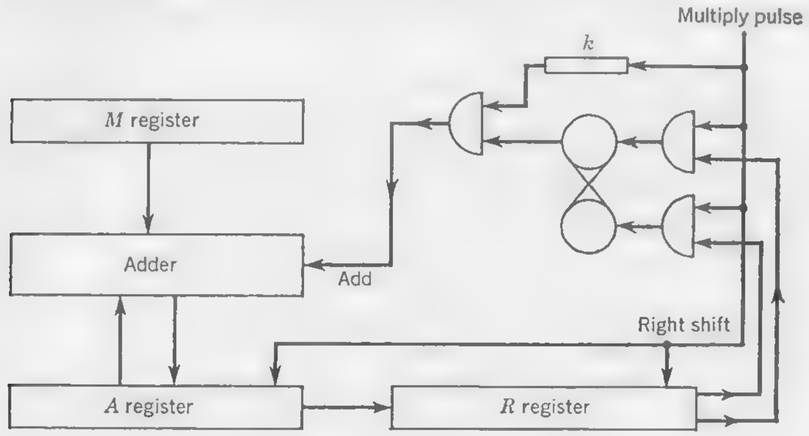


FIG. 15.17. Multiply control.

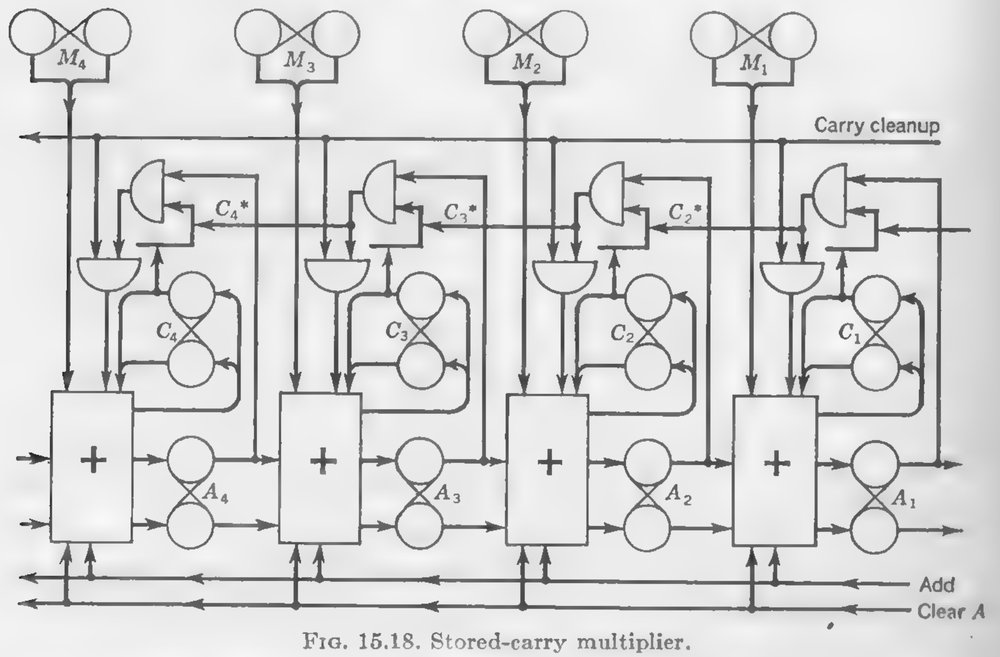


FIG. 15.18. Stored-carry multiplier.

allow for this carry and to allow for shifting the partial products and multiplier. In the computer with the carry time mentioned above, 8 μsec were allowed between additions so that multiplication by 36 binary bits required 288 μsec. A simplified diagram of such a multiplier appears in Fig. 15.17.

Another way of doing the multiplication does not allow time for carry propagation. Instead the carry is stored each time. This arrangement is shown in Fig. 15.18. Between each stage of the accumulator is a carry flip-flop which stores the carry. The output of the adder logic is not to the A register flip-flop of the same stage but to the next less significant stage; so there is no extra shift to be performed.

Table 15.8. Stored-carry Multiplication Example

M_4 0	M_3 1		M_2 1		M_1 0							After clock time
C_4	A_4	C_3	A_3	C_2	A_2	C_1	A_1	R_4	R_3	R_2	R_1	
0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0	0	0	1	1	1
0	0	0	1	1	0	0	1	0	0	0	1	2
0	0	0	1	1	1	0	0	1	0	0	0	3
0	0	0	0	1	0	0	1	0	1	0	0	4
0	0	0	0	0	1	0	0	1	0	1	0	5

Multiplicand in M..... 0110
Multiplier in R..... 0111
Answer in A and R..... 0010 1010

The operation of the various adders is defined by the following equations:

$$A_1s = M_1 \cdot C_1 \cdot A_2 + M_1 \cdot nC_1 \cdot nA_2 + nM_1 \cdot C_1 \cdot nA_2 + nM_1 \cdot nC_1 \cdot A_2$$
$$A_1r = nM_1 \cdot nC_1 \cdot nA_2 + nM_1 \cdot C_1 \cdot A_2 + M_1 \cdot nC_1 \cdot A_2 + M_1 \cdot C_1 \cdot nA_2$$
$$C_1s = M_1 \cdot A_2$$
$$C_1r = nM_1 \cdot nA_2$$
$$C_k* = (C_{k-1} + C_{k-1}*) \cdot A_{k-1}$$

After all the additions corresponding to the various 1 bits of the multiplier have been done, time is allowed for the final propagation of carry, and the carry cleanup pulse is issued. Circuitry which in the arrangement of Fig. 15.16 gives a carry time of 4 μsec can operate at clock frequencies of 2 Mc. Thus, not counting memory access for the operands, the whole multiplication could be done in less than 25 μsec. The multiplier digits would be processed in 18 μsec, and the time at the end is for the final propagation of the carry.

The shifting inherent in the design of Fig. 15.18 does not interfere with addition, since in that case the first operand is established in the carry flip-flops C, and the second in M. The first add pulse establishes part of the answer in the A register and the carries in the C register. A carry cleanup pulse produces the final answer in A in the right position. The first operand can be transferred to the C register by special gates, or the transfer can be accomplished by a pulse on the add line by first setting the A register to 1's. This circuit is not much more complicated than that of Fig. 15.16 and multiplies approximately ten times faster.

Note that the multipliers of Figs. 15.17 and 15.18 both shift the partial products to the right (in the less significant direction), whereas the multiplier of Sec. 15.14 shifted the partial products in the more significant direction. It is difficult to do division by trial subtraction in an accumulator using right shifts. If left shifts were used in the multiplication process then it would be necessary to have carry circuits for two full registers. In the serial system of Fig. 15.14 the extra carries required no extra circuitry, except perhaps minor changes in carry suppression. However, in a parallel system this extra carry represents a nontrivial amount of equipment. Therefore, one solution is to add left-shift circuits in order to accomplish division, and mechanize a system very similar to that described in Sec. 15.16.

15.18. SERIAL-PARALLEL ARRANGEMENTS

In order to gain a factor of speed over the serial systems without using all the equipment required in parallel systems, computers have been built using several channels processing groups of binary characters in parallel. The most frequent arrangement is four-channel systems.

The interconnection of the arithmetic registers in a four-channel system may still be essentially as illustrated in Fig. 15.17 where each path now carries four bits in parallel. The four-channel adder may be that shown in Fig. 15.16. At a given clock time 4 binary bits of the number in the accumulator register are established in the A flip-flops, and 4 bits (one hexadecimal digit) of the addend are established in the M flip-flops. A carry digit (if present) from the previous operation is established in the corrective-1 flip-flop (now perhaps best called the propagated carry flip-flop). During the clock time the carry propagates through four binary stages; at the next clock time the answer is sent out and the propagated carry is established in the carry flip-flop.

Negative numbers are processed by the use of complements. In this case it is a little more difficult to obtain the 2's (16's) complement by the scheme shown in Fig. 15.12. Usually a 1's (15's) complement is used, and a corrective-1 is inserted in the carry input to the units position.

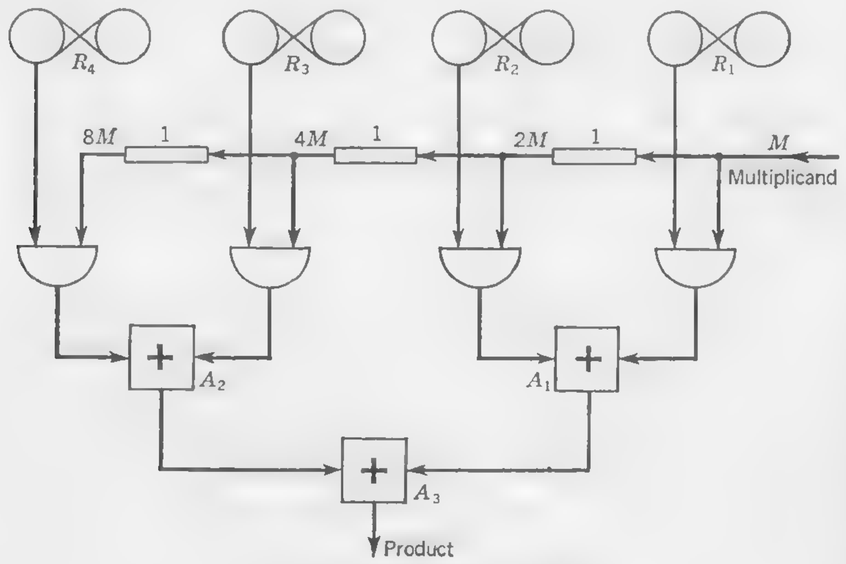


FIG. 15.19. Whiffletree multiplier.

Multiplication and division are more of a problem in four-channel systems. If simple binary operations are to be done then the registers must shift serial in a binary mode as well as in the four-channel mode. The alternative is to do multiplication and division in base 16 arithmetic. This requires more complex circuitry such as that described in the section on decimal arithmetic.

An alternate form of serial-parallel arrangement is illustrated in Figs. 15.19 and 15.20. Figure 15.19 shows part of a so-called "whiffletree" multiplier. The binary multiplicand arrives on a single channel and through appropriate delay lines two, four, and eight times the multiplicand are produced. As shown, four bits of the multiplier gate these factors and a series of single-channel binary adders combine these to give the product. In a complete system for a serial computer, multiplication (producing a two-word answer) is done in just two word times.

Figure 15.20 shows a multiplier which combines the whiffletree and the four-channel ideas. The multiplier shifts in such a way as to halve the individual hexadecimal digits. During each cycle the multiplicand is doubled by the adder A_1. The units bits of the digits of the multiplier ($R_1, R_2, \ldots, R_6$) control the addition of the multiplicand (or one, two, four, or eight times the multiplicand depending upon the cycle) by the adders $A_2, \ldots, A_7$. In four cycles all bits of each multiplier digit have been processed so the multiplication is complete. For binary numbers this scheme does not compete with the simple whiffletree of Fig. 15.19. If the adders $A_1, \ldots, A_7$

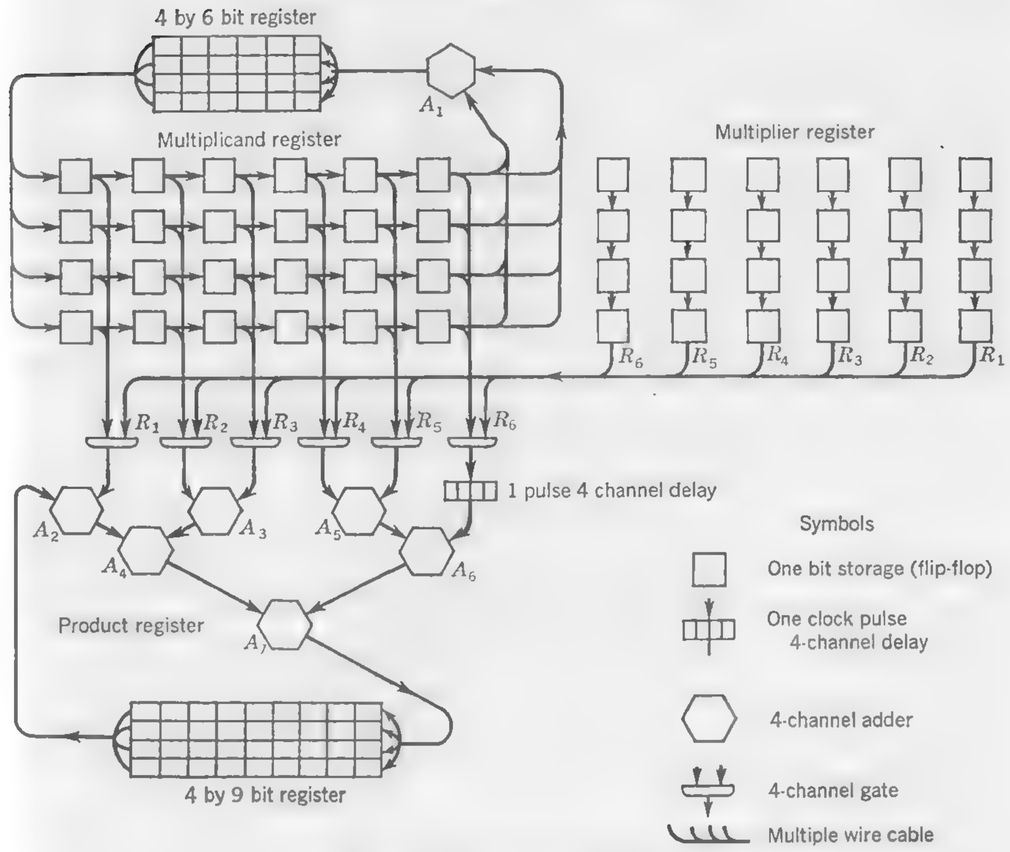


FIG. 15.20. A four-channel multiplier.

were decimal adders then the system could accomplish decimal multiplication in four cycles.

15.19. DECIMAL-ARITHMETIC OPERATIONS

15.19.1. Addition and Subtraction. As in the binary systems subtraction is usually handled by the use of complements with respect to 10^k or $10^k - 1$. In systems where the decimal digits are represented by binary numbers (see Sec. 15.5), usually a binary adder is used and the addition of two decimal digits occurs in two steps. First there is a normal addition of the two digits, then a possible corrective addition to account for the gap of six in the binary representation of the decimal digits. In the simple binary representation (1,2,4,8), six must be added if the sum of two digits is 10 or greater. In the excess-three system, three must be added or subtracted depending upon whether or not there is a decimal carry.

15.19.2. Multiplication and Division. In most decimal-arithmetic units the operations of multiplication and division are done by repeated addition or subtraction. In such schemes a multiplication of two 10-digit decimal numbers may require a maximum of 90 additions with the appropriate number of shifts. In order to speed up multiplication, arithmetic systems have been designed in such a way that, if a multiplier digit exceeds five, ten times the multiplicand is added (this might be accomplished by adding the multiplicand before shifting), then the appropriate number of subtractions is performed. In order to attain even greater speed, systems have been built in which one to nine times the divisor or multiplicand is provided and the whole multiplication or division becomes essentially equivalent to 10 additions or subtractions.

15.20. FLOATING-POINT NUMBERS

Most digital computers have been built to handle efficiently numbers consisting of the equivalent of from 5 to 15 decimal digits. In many computations this range is not sufficient to compute with adequate precision all the numbers that occur in a problem. If this range is not sufficient there must be a means of carrying an exponent for those numbers whose range is extreme. One method is to have the programmer keep track of these exponents and to scale each variable occurring in the computation in such a way that capacity is not exceeded and sufficient accuracy is obtained. Some computers have been built with floating-point arithmetic units which automatically keep track of this exponent and shift the significant figures in such a way as to provide maximum significance. In all general-purpose computers it is possible to program routines which do the equivalent process. However, these programming techniques slow down the computer by hundreds of times, and therefore, there is considerable premium on proper scaling and running of the problem in the fixed-point mode.

15.20.1. Representation of Floating-point Numbers. A floating-point number will be written in the following form:

$$n \times b^e \qquad \text{or} \qquad (n,e)$$

The significant figure n will be of fixed length (less than one word length) and will denote a rational number lying between plus and minus 1. The exponent e is an integer usually lying in the range of about -50 to 50. The base of the number system b is either 2 or 10. (Binary-arithmetic systems are simpler than decimal systems and therefore faster; so it now seems that there may always be binary-arithmetic systems in digital computers.)

The second representation given above is indicative of the form of storage used in computers; that is, the significant figures n and the exponent are stored in the same word in the computer. Each quantity may carry its own sign, and complementary or excess-50 exponents may be used. In a fixed-point computer the sign bit in the computer is usually assigned to n and the exponent is carried in excess-50 (or perhaps excess-64) form. Using excess-50, and writing the exponent ahead of the significant figures with a decimal point for separation purpose, some examples of floating-point numbers follow:

Number	Excess-50 equivalent	International equivalent
1	$51.10000 \cdots 0$	$.100 \cdots 0_{10}1$
-1	$-51.10000 \cdots 0$	$-.100 \cdots 0_{10}1$
$\frac{1}{3}$	$50.33333 \cdots 3$	$.333 \cdots 3_{10}0$
7,568,932.76	57.756893276	$.756893276_{10}7$

The third representation of decimal floating-point numbers is the accepted international algebraic language form (see Sec. 17.3) proposed by an international committee on programming language.

15.20.2. Addition of Floating-point Numbers, General Discussion. Let $(n1,e1)$ and $(n2,e2)$ represent two floating-point numbers to any base. If $e1$ and $e2$ are equal, the significant figures may be added directly. If they are not equal, then it is necessary to shift either $n1$ or $n2$ and adjust the exponent until equality is reached. The sum of $n1$ and $n2$ may have one more significant figure than either $n1$ or $n2$. In most computers this requires that the answer be shifted one position to the right, and the exponent increased by unity. The following examples illustrate the possibilities:

$31456_{10}5$	$31456_{10}5$	$31456_{10}5$	$31456_{10}5$
$51128_{10}5$	$51128_{10}3$	$00511_{10}5$	$94157_{10}5$
$82584_{10}5$	$31967_{10}5$	$31967_{10}5$	$12561_{10}6$

Subtraction is just a matter of changing the sign of the second operand and adding.

15.20.3. Representation of Zero. If n has five significant digits as in the illustrations above then five 0's with any exponent whatsoever represents 0. When one writes $.00000_{10}17$ all that is specified about the number is that it lies in the range from $-.000005_{10}17$ to $.000005_{10}17$. If no other information is given, presumably any value in the range is equally probable; that is, the distribution of possible values is rectangular. However, if a number such as that given above is the result of a computation, the distribution of the possible results is more apt to approximate a normal distribution (see the central-limit theorem in statistics).

If the range in the exponent is from -50 to 50 then for most purposes $.00001_{10} - 50$ is a satisfactory representation of zero. Thus, it is possible to devise a practical floating-point system in which zero is not used. On the other hand, an absolute zero is useful for logical purposes, and a number of systems have been used in which all 0's are written with the most negative exponent. Thus, any number subtracted from itself gives $.00000_{10} - 50$ in such a system.

15.20.4. Normalization. In most computing problems it is desirable to obtain as many significant figures in the answers as possible. In floating-point systems this can be accomplished by normalizing the nonzero numbers at each step. A nonzero number is in normal form if the first digit of n is not zero. If two numbers are subtracted in which the first digits are the same and they carry the same exponent, the result will have 0's in the first positions. Normalization will shift the result to the left and adjust the exponent. This has the effect of bringing in nonsignificant digits on the right, and after a lengthy computation relatively few (or even none) of the digits may be valid.

A process which tends to maintain the correct number of significant digits is the following: (1) On addition never shift results to the left; shift right to recover overflow. (2) On multiplication normalize the factor whose number n is greatest in absolute value. (3) On division normalize the denominator. A rule for division similar to that for multiplication would be better, but the rule given above is easier to use in most arithmetic circuits.

15.20.5. Floating-point Addition. The process of floating-point addition is described in flow-diagram form. The numbers in parentheses to the right correspond to the comments about the process given below.

Assume that the two operands are $(n1,e1)$ and $(n2,e2)$. The flow diagram of Fig. 15.21 can be considered as a programming flow diagram and used as a basis of a

1. $e1 - e2 \rightarrow e$. Test e:
   - $e > 0$: Shift $n2$ e places to right. $e1 \rightarrow e$. To 2. (1)
   - $e < 0$: Shift $n1$ e places to the right. $e2 \rightarrow e$. To 2.
   - $e = 0$: $e1 \rightarrow e$. To 2.
2. $n1 + n2 \rightarrow n$. Overflow?
   - Yes: $n1/10 \rightarrow n1$. $n2/10 \rightarrow n2$. $e + 1 \rightarrow e$. Overflow? (2)
     - Yes: Alarm. Halt. (3)
     - No: To 2.
   - No: Normalize and adjust e. Assemble n and e.

FIG. 15.21. Floating-point-addition flow diagram.

subroutine for use in a fixed-point computer, or it can be interpreted as the design diagram for the hardware to do floating-point addition.

1. A more elaborate test might be made here. If e is greater in absolute value than the number of places in n then there is no need to perform an addition since one of the operands is the answer. The present flow diagram is set up on the basis that shifts beyond the end of the n registers result in the digits being lost.

2. This line assumes the number system is decimal. If the overflow digit is not lost during addition an alternate procedure would be to shift the answer right one place and adjust the exponent.

3. This condition occurs when the exponent exceeds 50. Underflow occurs when the exponent is less than -50. Most systems substitute 0 for the answer in this case.

15.20.6. Floating-point Multiplication. Figure 15.22 shows the process of floating-point multiplication. The process of normalization which was only indicated in Fig. 15.21 is shown in detail here.

1. The multiplication of $n1$ by $n2$ should give a rounded-off answer with the same number of digits and scaled with the decimal point (or binary point) on the left.
2. This underflow test should also be made in addition.
3. This is the normalization test. That is, is the first digit of n not equal to zero? Note that if the first digit of n is zero then n is tested for zero.
4. Each time the exponent is reduced there is a possibility of underflow.
5. After normalization the two parts of the number are assembled into the answer register and the process is finished.

---

1. $n1 \cdot n2 \rightarrow n$, $e1 + e2 \rightarrow e$. overflow? (1)
   Yes: Alarm, Halt.
2. No: Underflow? (e less than -50)
3. Yes: $.00000 \rightarrow n$, $-50 \rightarrow e$, to 4. (2)
   No: $|n| \geq 100000$? (3)
   Yes: to 4.
   No: $n = 0$?
   Yes: to 3.
   No: $10n \rightarrow n$, $e - 1 \rightarrow e$, to 2. (4)
4. Assemble n and e, exit. (5)

FIG. 15.22. Flow diagram for floating-point multiplication.

15.20.7. Floating-point Speeds. The large factor of decrease in speed of programmed floating operations is apparent from inspection of the details of the flow diagrams in Figs. 15.21 or 15.22, since a single add command in fixed-point mode is replaced by a whole routine. In built-in floating point the situation is different. For example, multiplication may even be faster since it is not necessary to multiply by all the digits of a word, and usually there will be no shifting at all. For addition the time for the floating-point operation depends strongly upon the speed of shifting in the arithmetic registers. Note that, if one of the operands must be shifted before the addition, then the result will usually not have to be shifted. In practical problems the exponents of operands tend to be of comparable size; so the average amount of shifting is relatively small. More precise statements can be given only if more explicit assumptions about the operands are made.

15.21. STRUCTURE OF ARITHMETIC UNITS

Two extremes in organization of arithmetic units will be presented here. Both of these will be discussed from the point of view of the user; so there may be other registers present in any actual embodiment of the devices.

Figure 15.23 shows the structure of a simple arithmetic unit. It contains an accumulator, and the results of all arithmetic operations are left in the accumulator. It is not important in this discussion whether the numbers are in true or complementary form.

If (A) denotes the contents of the accumulator register and (a) denotes the contents of an address "a" in the memory and $\rightarrow$ is read as "replaces," the following relations describe the functions of the arithmetic unit.

1. $(a) \rightarrow (A)$
2. $(A) \rightarrow (a)$
3. $-(a) \rightarrow (A)$
4. $(A) + (a) \rightarrow (A)$
5. $(A) - (a) \rightarrow (A)$
6. $(A) \cdot (a) \rightarrow (A)$ (fractional)
7. $(A)/(a) \rightarrow (A)$ (provided $|(a)| > |(A)|$)

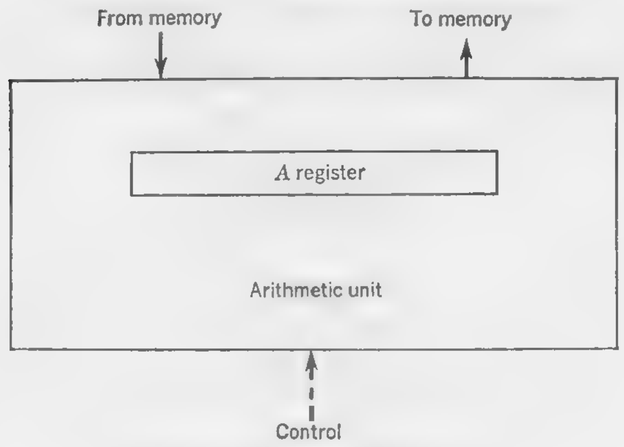


FIG. 15.23. Simple arithmetic unit.

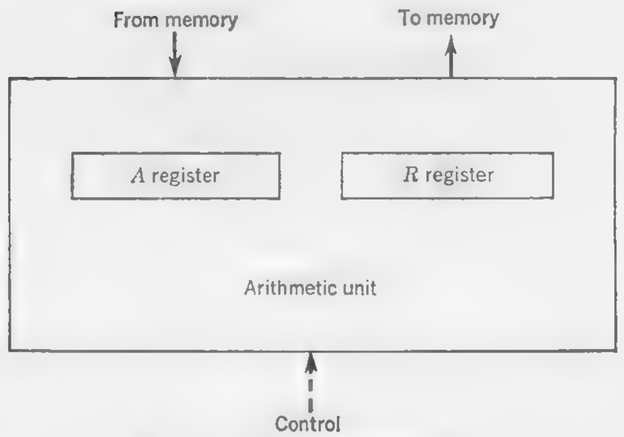


FIG. 15.24. A-R arithmetic unit.

Figure 15.24 shows a more complicated arithmetic unit. In this unit there is an accumulator A and a multiplier register R. The following relations describe the functions of this arithmetic unit:

1. $(a) \rightarrow (A)$
2. $(a) \rightarrow (R)$
3. $-(a) \rightarrow (A)$
4. $-(a) \rightarrow (R)$
5. $(A) + (a) \rightarrow (A)$
6. $(A) - (a) \rightarrow (A)$
7. $(R) \cdot (a) \rightarrow (R)$
8. $(R)/(a) \rightarrow (R)$
9. $(A) + (R) \cdot (a) \rightarrow (A)$
10. $(A) + (R)/(a) \rightarrow (A)$
11. $(A) \rightarrow (a)$

To compare the two types of arithmetic units consider the coding to evaluate the formula $a + b \cdot c + d \cdot e \rightarrow z$.

First Arithmetic Unit	Second Arithmetic Unit
1. $b \rightarrow (A)$	1. $a \rightarrow (A)$
2. $(A) \cdot c \rightarrow (A)$	2. $b \rightarrow (R)$
3. $(A) \rightarrow (t)$	3. $(R) \cdot c + (A) \rightarrow (A)$
4. $d \rightarrow (A)$	4. $d \rightarrow (R)$
5. $(A) \cdot e \rightarrow (A)$	5. $(R) \cdot e + (A) \rightarrow (A)$
6. $(A) + (t) \rightarrow (A)$	
7. $(A) + a \rightarrow (A)$	

In writing the above steps for the first arithmetic unit advantage was taken of the fact that the coder can scan the formula and not add in the "a" until at the end. In the second arithmetic unit the formula can be processed from left to right without intermediate storage of partial results.

The advantage in the second unit lies in the fact not only that the code is shorter, but also that no scanning is required, which means that there is less chance for mistakes.

References

1. Gilchrist, B., J. H. Pomerene, and S. Y. Wong: Fast Carry Logic for Digital Computers, *IRE Trans. on Electronic Computers*, **EC-4**(4): 133–136 (December, 1955).
2. Horton, J. W., and A. G. Anderson: A Full Binary Adder Employing Two Negative-resistance Diodes, *IBM J. Research Development*, **2**(3): 223–231 (July, 1958).
3. Robertson, J. E.: Two's Complement Multiplication in Binary Parallel Digital Computers, *IRE Trans. on Electronic Computers*, **EC-4**(3): 118–119 (September, 1955).
4. Ross, H. D., Jr.: The Arithmetic Element of the IBM Type 701 Computer, *Proc. IRE*, **41**:1287–1294 (October, 1953).
5. Torrey, R. D., and T. H. Bonn: A 2.5-megacycle Ferractor Accumulator, *Proc. Eastern Joint Computer Conf.*, New York, **T-92**:50–53, Dec. 10–12, 1956.
6. Weinberger, A., and J. L. Smith: A One-microsecond Adder Using One-megacycle Circuitry, *IRE Trans. on Electronic Computers*, **EC-5**(2): 65–73 (June, 1956).

Section 16

DIGITAL-COMPUTER-SYSTEM DESIGN

By

WERNER BUCHHOLZ, *Senior Systems Planner, International Business Machines Corporation, Poughkeepsie, N.Y.*

WILLIAM F. GUNNING, *Vice-President—Technical Director, Epsco-West, Inc., Anaheim, Calif.*

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

RAGNAR THORENSEN, *Director, Magnavox Research Laboratories, Torrance, Calif.*

CONTENTS

16.1. Sequencing of Computer Operations 16–2
16.2. Internal and External Program Control 16–2
16.3. Plugboard Control 16–2
16.3.1. Plugboard Hubs 16–2
16.3.2. Sequencing 16–3
16.3.3. Storage Addressing 16–5
16.3.4. Card Programming 16–5
16.3.5. Stored-program Control via Plugboard 16–5
16.4. Input and Output Control . . . 16–5
16.4.1. Buffer Storage and Translation 16–5
16.4.2. Selection and Interlocking 16–5
16.4.3. Data Rearrangement 16–6
16.5. Manual Controls 16–7
16.6. Structure of Stored-program Computers 16–7
16.7. A Single-address Stored-program Computer 16–8
16.8. Two-address Computers 16–9
16.9. Three-address Computers . . 16–10
16.10. Four-address Systems 16–10
16.11. A Single-address System with Index Registers 16–11
16.12. Elementary Logical Circuits . 16–11
16.13. A Memory Unit 16–12
16.14. Parallel and Serial Representation of Information 16–13
16.15. Periodicity of Information Flow 16–14
16.16. Translation of Information from Serial to Parallel Representation and Vice Versa . . 16–14
16.17. Timing and Synchronization of Memories 16–14
16.17.1. General 16–14
16.17.2. Influence of Physical Characteristics of Memories on Timing and Synchronization 16–14
16.17.3. Periodicity Imposed through Regeneration . . . 16–15
16.17.4. Periodicity Imposed through Scanning and Selection 16–15
16.17.5. Access Time and Information Transfer 16–15
16.18. Transfer of Information between Memories of Different Capabilities 16–16
16.19. Timing and Synchronization of Arithmetic Elements (General) 16–17
16.20. Serial Arithmetic-element Timing and Synchronization . 16–18
16.21. Parallel Arithmetic-element Timing and Synchronization . 16–20

16.22. Transfer and Shifting in Double Registers (Static Primary Storage)..... 16–21
16.23. Transfer and Shifting in Single Registers (Static Primary Storage)..... 16–22
16.24. Transfer and Shifting (Dynamic or Regenerated Primary Storage)..... 16–23
16.25. Transfer and Shifting (Delay-line Primary Storage)..... 16–24
16.26. Comparison of Shift-register Methods—A-C vs. D-C Coupling..... 16–24
16.27. Timing of the Carry Operation..... 16–25
16.28. Timing and Synchronization in the Control Section—Introduction..... 16–28
16.29. Sequencing of Instructions.. 16–28
16.30. Sequencing of Commands Synchronously (Fixed Minor Cycle)..... 16–30
16.31. Sequencing of Commands Synchronously (Variable Minor Cycle)..... 16–31
16.32. Sequencing of Commands Asynchronously (Self-timing)..... 16–32
16.33. Microprogramming..... 16–32

16.1. SEQUENCING OF COMPUTER OPERATIONS

A characteristic feature of electronic digital computers is the ability to perform a sequence of operations. The sequence of operations is under the control of a *program* which is set up in advance of the time when the computer is to perform the desired functions. The portion of a computer concerned with the interpretation of a program and with the execution of the various steps of the operations called for by the program is commonly called the *program control.* It ties together the arithmetic, storage, and input-output sections of a computer to make them operate as a whole. By varying the sequence of operations, the program control performs much of the logic needed.

This section is concerned with the general nature of the program-control equipment found in many digital computers. The techniques for programming, or coding, computers are discussed in Sec. 17.

16.2. INTERNAL AND EXTERNAL PROGRAM CONTROL

There are two basically different methods of storing the program. The storage device may be internal to the computer and not directly accessible to the operator: either the main data storage or some auxiliary program storage. Such a machine is generally referred to as a *stored-program* computer.

In other machines the program-storage device is external and accessible to the operator. The program may be set up on switches, punched cards, pins, or wired panels. The most common method consists of pluggable wires on removable panels, variously known as *plugboards, control panels,* or *patch panels.*

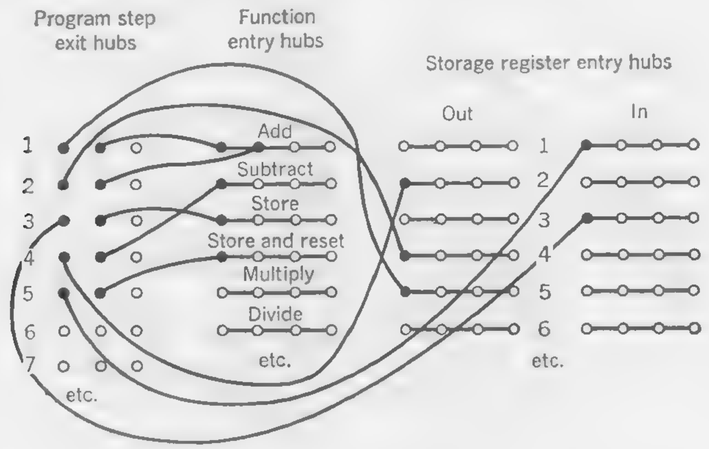
Although these are all forms of storage, the external devices are less flexible than internal storage devices since they cannot be set up or changed much under computer control. Hence the term *stored program* is reserved for a program stored internally on the same medium (magnetic drums, magnetic cores, delay lines, electrostatic storage, etc.) as is used for data. Large high-speed computers generally use the stored-program approach, whereas smaller machines frequently employ plugboards. In addition, plugboards may be found on input-output devices for either type of machine.

Both stored-program and plugboard techniques will be illustrated because of their widely different characteristics.

16.3. PLUGBOARD CONTROL

16.3.1. Plugboard Hubs. A typical plugboard consists of an insulating bakelite panel with regularly spaced holes or *hubs.* Wires with special contact tips are plugged into appropriate pairs of holes; when a wired plugboard is inserted into the corresponding frame on the computer, the tips make connection with stationary contacts to complete the various circuits.

Figure 16.1 shows some typical plugboard hubs in schematic form. Some of the hubs are *exits* for emitting control pulses; others are *entries* for receiving these pulses. Wires are normally plugged from an exit hub to an entry hub. The exit hubs shown correspond to the various steps of a program; in the example it is assumed that program steps are energized sequentially starting with step 1. At each step three independent exit hubs are shown to permit wiring to up to three entry hubs. One program exit hub is usually wired to a function entry hub, such as ADD (or READ IN +), SUBTRACT (or READ IN −), STORE (or READ OUT), MULTIPLY, etc. Another program



Problem shown wired: $A+B=C$
$C-D=E$

Storage register assignment:

Data	Register
A	5
B	4
C	3
D	2
E	1

FIG. 16.1. Typical plugboard hubs.

exit is plugged to the storage register to be referred to, either coming out of or going into the register. A third exit may be wired to control shifting (not shown).

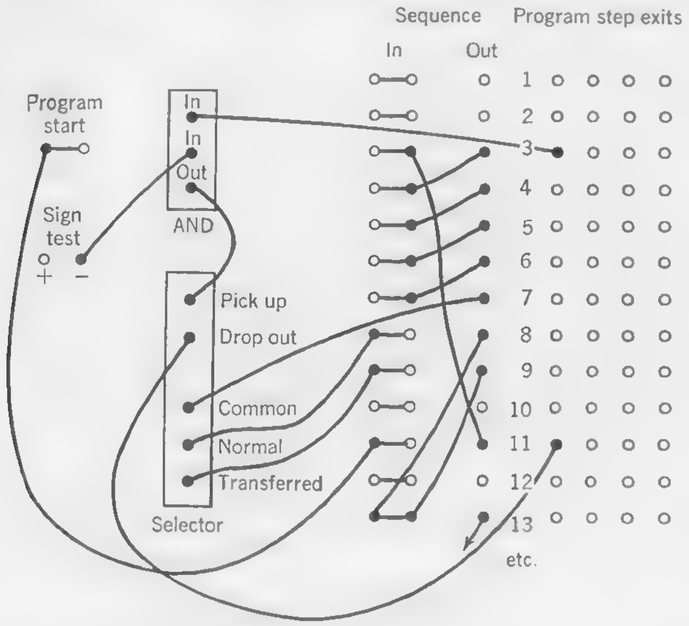
Several hubs are shown joined by straight lines; these are to indicate permanent connnections which facilitate connecting more than one wire to the same electrical point without resorting to multiple connectors. In Fig. 16.1, wires go to ADD from two different program steps.

The three independent exit hubs at each program step each connect to a separate electronic driving circuit. Separate drivers are used as isolation to avoid *back circuits* which would produce undesired coupling between entry hubs. In Fig. 16.1, the two wires going to ADD from steps 1 and 2 would cause both registers 4 and 5 to operate on either step if the exits were not isolated. Design precautions are also needed to avoid overloading of circuits which might distort the pulses and cause erratic operation.

16.3.2. Sequencing. There are various methods in use for the important process of controlling the sequence of program steps. An elementary method is to wire all the steps which might be needed and then, by means of special SUPPRESS hubs, suppress the operation of undesired individual steps under the control of the preceding program. A variation is to turn on a SKIP function at one program step and to turn it off at some subsequent step, all intermediate steps being skipped through at high

speed. In either case, program steps are operated sequentially regardless of whether anything is wired to function on each step. The basic driving circuit is a ring counter or equivalent. Iteration loops can be wired by causing the ring to repeat its sequential stepping as many times as necessary.

A more flexible but also more costly arrangement is to equip each program step with its own sequencing circuit which permits it to be turned on and off independently. In Fig. 16.2, each program step is provided with IN and OUT sequence hubs. To begin



Sequence as wired:

Start	
Step 11	Reset selector to normal
Step 3	Pick up selector if sign is minus
Step 4-6	
Step 7	Go to step 8 if sign was plus on step 3, to 9 if minus
Step 8 or 9	Go to step 13 from either step
Step 13	

FIG. 16.2. Program steps with arbitrary sequencing.

with, a START pulse is wired to the IN of any desired step, its OUT is wired to the IN of the next step, and so on, regardless of the way the steps are labeled. Thus the sequence can be quite arbitrary.

Figure 16.2 also shows how the sequence can be altered selectively. One or more *selectors* are provided which are individual flip-flops with setting and driving circuits brought out to the plugboard in a manner resembling a single-pole double-throw latch relay. A circuit is established from the *common* to the *transferred* hub when the selector is *picked up*, and from the *common* to the *normal* hub when the selector is *dropped out*. The selector can be set according to some condition at one program step and used to control sequence at some later step. In Fig. 16.2, a selector is dropped out (reset) on step 11, picked up on step 3 if the sign is minus (via an AND circuit which produces an output only if both inputs are energized), and later used at the end of step 7 to select either step 8 or step 9 depending on the sign at step 3. The program then proceeds to step 13 from either step 8 or step 9.

16.3.3. Storage Addressing. In Fig. 16.1, the proper storage register is selected at each step by plugging a wire to a hub directly corresponding to the desired storage register.

This direct method is not practical when the number of storage positions becomes large. Instead, numerical addresses are given to the storage positions and a storage location is selected by inserting the corresponding address into an address register, of which there may be several.

The plugboard then has a few hubs corresponding to each address register to cause reading into or out of storage according to the address in this register, or to enter a new address into the register, or to add a constant to the register so as to advance to the next address.

By means of suitable interconnections, the address registers may also be used as index registers.

16.3.4. Card Programming. Instead of having a fixed program wired on the plugboard, it is possible to wire up several independent short programs and to have the input device select which program to use at any one time. For example, a deck of punched cards may be entered, each card bearing a set of data and the operation FLOATING-POINT ADD, another may have a number and SQUARE ROOT. The operation codes punched in the cards are decoded in an electronic or relay matrix, known as a *digit selector*, so as to select the proper program step at which to start.

16.3.5. Stored-program Control via Plugboard. A given plugboard is limited by the number of program steps available, and an elaborate problem may exceed the limits of a plugboard. It may be possible to break down such a problem into sections, each of which will fit a plugboard. Another possibility is to supplement the plugboard with a stored program.

A plugboard computer equipped with address registers and digit selectors can be set up to operate with stored instructions as well. Every distinct operation is set up on one or more program steps of the plugboard which direct the computer to perform the corresponding operation. One of the address registers is wired as an instruction counter to address the storage locations containing the instructions. By means of plugboard wiring, the address digits of the instructions are sent to other address registers, and the operation code digits are used to select the proper starting point on the plugboard corresponding to each operation.

16.4. INPUT AND OUTPUT CONTROL

16.4.1. Buffer Storage and Translation. Input and output units may communicate directly with the main storage, or the data may flow through an intermediate *buffer storage*. Such a buffer generally requires separate addressing and control circuits, so as to permit data to flow in from an input unit or flow out to an output unit independently of the program in the main computer.

Translation circuits are needed whenever the data representation on the input-output media differs from the code used inside the computer. Another possible function of translation and buffer storage circuits is to change from serial to parallel representation, or vice versa.

16.4.2. Selection and Interlocking. In a computer with multiple input and output units, two addresses must be specified, one to select the desired unit and the other to specify the storage area for the data to be read or written. This may be done by one multiple-address instruction or two single-address instructions in succession. Where the amount of information transferred at one time is not fixed by design, the number of words or characters must also be given, either as a count or by a special mark placed at the end of the block of information to be read or written.

Input-output devices may not always be available when the computer program calls for them. Tapes have to be changed, printers occasionally require a new supply of paper, and card machines can run out of cards. Interlock circuits are thus provided to hold up the computer or the input-output unit when the other is not ready. When buffer storage is involved, two sets of interlocks are needed: one to synchronize the

buffer storage with the input-output unit, and the other to keep the buffer storage in step with the computer.

16.4.3. Data Rearrangement. Both input and output data frequently require a considerable amount of detailed rearranging, and special provisions may be made to facilitate this. A typical problem occurs in printing numbers when it is desired to suppress zeros at the left of a number and to insert decimal point and commas.

Figure 16.3 gives an example. For easier reading it is necessary to be able to spread out the numbers in an arbitrary columnar arrangement and to spread individual

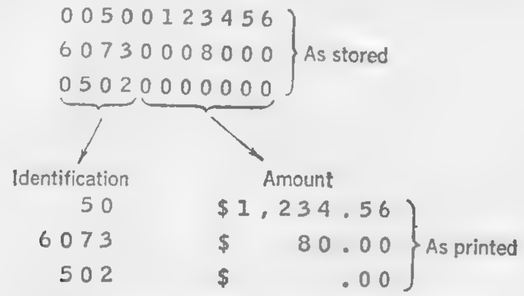


FIG. 16.3. Example of arranging for printing.

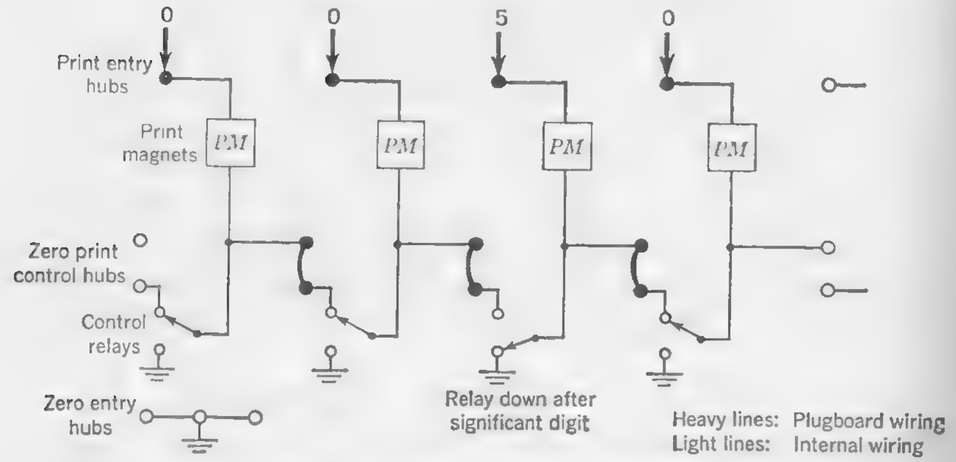


FIG. 16.4. Zero control wiring on plugboard.

digits for insertion of symbols. Generally it is desirable to suppress printing of all nonsignificant zeros, as shown in the left column. In the right column the insertion of a comma among nonsignificant zeros is also shown, but nonsignificant zeros to the right of the decimal point may have to be preserved as in the bottom line.

There are several techniques in use for controlling printing. One is to have a plugboard on the printer which has the ability to control individual characters. Figure 16.4 shows a method for suppressing zeros. Numeric digits are represented by timed pulses entering the *print entry* hubs at the top and passing through *print magnets* and through points of control relays. The magnets operate only if a connection is made to ground via the relay points. Digits 1 to 9 print before digit 0, and all relay points are down. At 0 time those relays corresponding to positions where no digit 1 to 9 has occurred are transferred to the upper contact. A pair of *zero print control* hubs is shown jumpered for each digit position except the highest. As a result, zeros to the right of a significant digit find a return path via the contact of that digit, and they are printed; zeros to the left are without a return path and cannot print.

Alternatively, if all zeros are to print, the bottom *zero print control* hubs are all wired to *zero entry*, which provides the necessary path to ground. Analogous wiring can be used for controlling decimal points, commas, etc.

An electronic method of suppressing zeros is to inspect each digit of a number serially, starting at the left. As long as only zeros are encountered they are changed to "blank" symbols which will not print, but the first significant digit is noted and all subsequent zeros are printed. If this inspection occurs during a serial transfer from one register to another, the digits can also be spread to permit insertion of columnar spaces and punctuation symbols. These operations can be put under the control of a stored program if means are provided for specifying the number of digits to be transferred and where the extra symbols are to be inserted.

16.5. MANUAL CONTROLS

A few basic manual program controls must be provided with any computer to accomplish

1. Loading the program (if stored) or inserting a plugboard
2. Setting the instruction counter or program step to the starting point
3. Starting the program

A large variety of additional manual program controls are possible, depending on the design. Among the common ones are buttons or switches to

4. Interrupt the program
5. Set the instruction counter to any arbitrary address
6. Execute single instructions
7. Operate the program step by step
8. Alter the program by controlling conditional transfers in the program depending on the switch settings
9. Stop the program at conditional stop instructions
10. Reset all conditional flip-flops.

Displays, such as neon lights, may be provided for the program to signal the operator under various conditions.

16.6. STRUCTURE OF STORED-PROGRAM COMPUTERS

As indicated in Fig. 16.5 stored-program computers consist of a memory, arithmetic unit, and a control unit. Input and output will not be considered now even though in many cases they may represent a substantial fraction of the whole computing system.

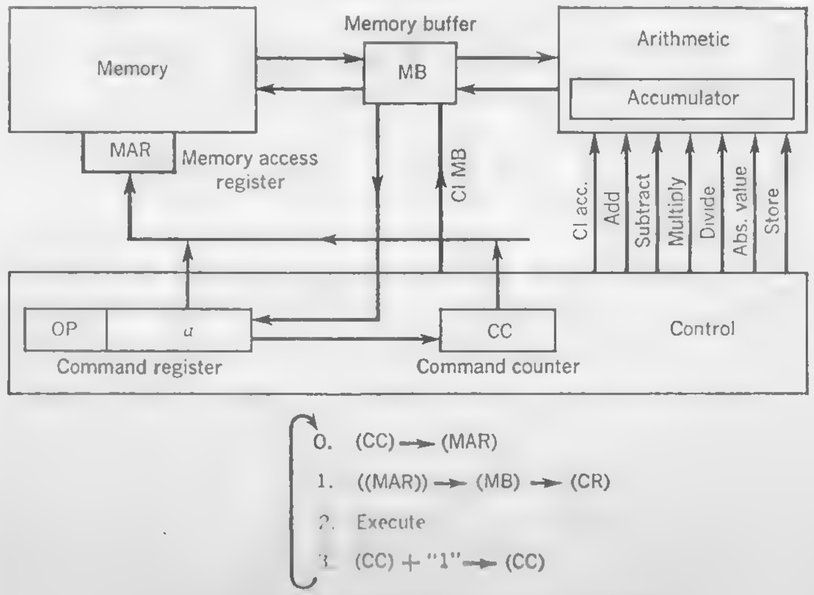


FIG. 16.5. A single-address computing system.

Suppose that the memory consists of 10,000 locations numbered from 0000 to 9999. Transfer of information to or from the memory is via the memory buffer *MB*. The particular location involved in the transfer is determined by the number in the memory access register *MAR* (a number between 0000 and 9999, inclusive).

The arithmetic circuits include an accumulator where the results of the various arithmetic operations will be developed. Generally, the operands for an arithmetic operation will be in the accumulator and in the memory buffer *MB*. The result will be in the accumulator *Acc*. In the following discussion it is not important whether the arithmetic is done in the binary or in the decimal system and whether the operations are floating-point or fixed-point.

The control circuits always include a command register *CR* and sometimes a command counter *CC*. Some of the possible arithmetic operations are indicated in Fig. 16.5 by the connections from the control unit to the arithmetic unit.

Notation. If *R* is the name of a register then (*R*) will be used to denote the number or unit of information stored in that register. Thus, in terms of Fig. 16.5, (*MB*) is the number stored in the memory buffer, (*Acc*) is the number in the accumulator, and (*MAR*) is the number (between 0000 and 9999) stored in the memory access register *MAR*. If (*MAR*) = 0176 then ((*MAR*)) = (0176) is the unit of information stored in location 176 in the memory. The symbol "→" will be read as "replaces." Thus, a transfer from one register to another can be written as (*MB*) → (*CR*). The operation of addition can be written as (*MB*) + (*Acc*) → (*Acc*). The statement (*CC*) + "1" → (*CC*) stands for the process of adding unity to the contents of the command counter.

16.7. A SINGLE-ADDRESS STORED-PROGRAM COMPUTER

The general sequence of events in the operation of a single-address computer is shown on the bottom of Fig. 16.5. In step 0 the content of the command counter is transferred to become the content of the memory access register. In step 2 the memory is accessed for an item which is transferred to the memory buffer and to the command register. This quantity is a command which determines what happens in the computation. In step 2 this command will be executed, and it may cause operations such as addition, subtraction, or multiplication. All the illustrated operations refer to the arithmetic unit and are shown by arrows extending from the control unit to the arithmetic unit in Fig. 16.5. Input and output commands are not discussed here.

After the execution of the command the contents of the command counter is increased by 1 and the whole process goes back to step 0 and repeats.

A more complete description of the operation of the device is shown in the computing-system flow diagram of Fig. 16.6. Here, again, steps 0, 1, 2, and 3 are shown, except that step 2 is shown in detail.

0. (CC) → (MAR),
1. ((MAR)) → (MB) → (CR),
2. What is OP?
   (*a*) AD (Add): a → (MAR), (a) → (MB), (MB) + (Acc) → (Acc), to 3.
   (*b*) SU (Subtract): a → (MAR), (a) → (MB), (Acc) − (MB) → (Acc), to 3.
   (*c*) CA (Clear and add): "0" → (Acc), to 2(a).
   (*d*) CS (Clear and subtract): "0" → (Acc), to 2(b).
   (*e*) MU (Multiply): a → (MAR), (a) → (MB), (MB) · (Acc) → (Acc) (properly rounded), to 3.
   (*f*) DI (Divide): a → (MAR), (a) → (MB), (Acc)/(MB) → (Acc), to 3.
   (*g*) AV (Absolute value): |(Acc)| → (Acc), to 3.
   (*h*) ST (Store): a → (MAR), (Acc) → (MB) → ((MAR)), to 3.
   (*i*) TC (Transfer control): a → (CC), to 0.
   (*j*) TN (Transfer on negative): { (Acc) < "0": to 2(i). (Acc) ≥ "0": to 3.
3. (CC) + "1" → (CC), to 0.

FIG. 16.6. A single-address computing system—flow diagram.

16.8. TWO-ADDRESS COMPUTERS

Figures 16.7 and 16.8 show some of the details of a two-address computer. In contrast to the computer of Fig. 16.5 where each command had a single address *a*, here each command has two addresses *a* and *b*. In this system the address *b* specifies the location of the next command and thus eliminates the need for the command counter *CC*. This type of system is of advantage in computers using cyclic memories in that commands can be located in positions such that they are available upon the completion of the preceding operation. Note that no transfer of control command is needed since every command transfers control.

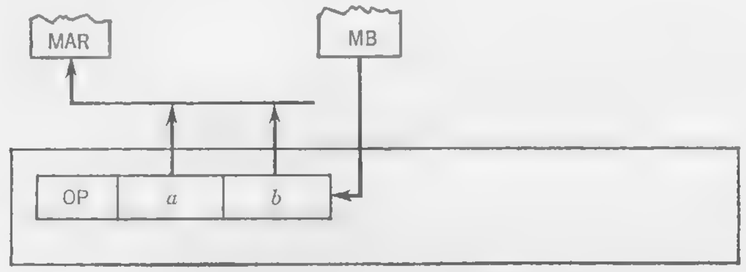


FIG. 16.7. Control for a two-address computer.

0. b → (MAR),
1. ((MAR)) → (MB) → (CR),
2. What is OP?
   (*a*) AD (add): a → (MAR), (a) → (MB), (MB) + (Acc) → (Acc), to 3.
   (*b*) · · ·
   · · ·
   (*j*) TN (transfer on negative): { (Acc) < 0: a → (MAR), to 1. (ACC) ≥ 0: to 0.
3. (CC) + "1" → (CC), to 0.

FIG. 16.8. A two-address computing system.

0. (CC) → (MAR),
1. ((MAR)) → (MB) → (CR),
2. What is OP?
   (*a*) AD (add): a → (MAR), (a) → (MB) → (Acc), b → (MAR), (b) → (MB), (MB) + (Acc) → (Acc) → (b), to 3.
   (*b*) · · ·
   · · ·
   (*j*) TN: { (Acc) < 0: a → (CC), to 0. (Acc) > 0: to 3. (Acc) = 0: b → (CC), to 0.
3. (CC) + "1" → (CC), to 0.

FIG. 16.9. Another two-address computing system.

Figure 16.9 gives the structure in flow-diagram form of a two-address system where the second address specifies an operand and result location. Here the command counter has to be used to specify the location of the next command. In this system each address in the memory plays the role of an accumulator. Note that the transfer on negative (*TN*) command has really become a "test" command in that it distinguishes three possibilities, negative, zero, and positive.

16.9. THREE-ADDRESS COMPUTERS

Figure 16.10 shows a three-address computer in flow-diagram form. The essential difference between this one and the computer of Fig. 16.9 is that the locations of the

0. (CC) → (MAR),
1. ((MAR)) → (MB) → (CR),
2. What is OP?
   (*a*) AD: a → (MAR), (a) → (MB) → (Acc), b → (MAR), (b) → (MB), Add, to 3.
   (*b*) · · ·
   · · ·
   (*j*) TN: { (Acc) < 0: a → (CC), to 0.
   (Acc) = 0: b → (CC), to 0.
   (Acc) > 0: c → (CC), to 0.
3. c → (MAR), (Acc) → (MB), Record, to 4.
4. (CC) + "1" → (CC), to 0.

FIG. 16.10. Three-address computing system.

two operands and the result are entirely independent in Fig. 16.10, whereas in Fig. 16.9 one of the operands and the result must be in the same location.

16.10. FOUR-ADDRESS SYSTEMS

Figures 16.11 and 16.12 describe a four-address system. Here the arithmetic operations are similar to those in the three-address systems. However, the fourth address can be used in case of "overflow" in the arithmetic circuits, as indicated in step 3 of Fig. 16.12.

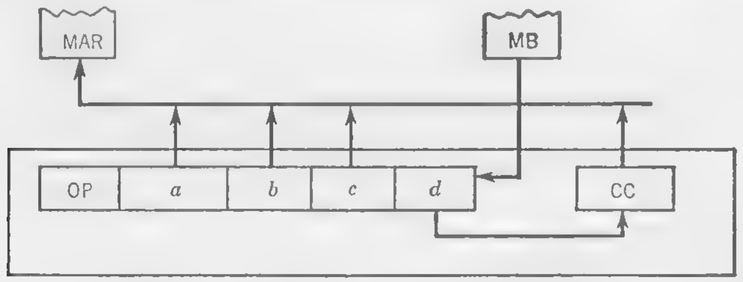


FIG. 16.11. Control for a four-address computer.

0. (CC) → (MAR),
1. ((MAR)) → (MB) → (CR),
2. What is OP?
   (*a*) AD: a → (MAR), (a) → (MB) → (Acc), b → (MAR), (b) → (MB), (MB) + (Acc) → (Acc), to 3.
   (*b*) · · ·
   · · ·
   (*j*) CO(Compare): a → (MAR), (a) → (MB) → (Acc),
   b → (MAR), (b) → (MB), (Acc) − (MB) → (Acc), Store,
   { (Acc) < 0: d → (CC), to 0.
   (Acc) ≥ 0: to 4.
3. Store, { Overflow: d → (CC), to 0.
   No Overflow: to 4.
4. (CC) + "1" → (CC), to 0.

Store: c → (MAR), (Acc) → (MB) → ((MAR)), Return.

FIG. 16.12. Four-address computing system.

Note that the STORE procedure given at the bottom of Fig. 16.12 is used in several places in the main flow diagram (see steps 2*j* and 3, for example).

The design flow diagrams (Figs. 16.6, 16.8, 16.9, 16.10, and 16.12) are very similar to flow diagrams used in programming (see Sec. 17) and the use of a subprocedure such as STORE (Fig. 16.12) is similar to the use of subroutines or subprograms.

16.11. A SINGLE-ADDRESS SYSTEM WITH INDEX REGISTERS

Figures 16.13 and 16.14 show the essential features of this system. Note that the control circuits now have an adder which is capable of adding two addresses.

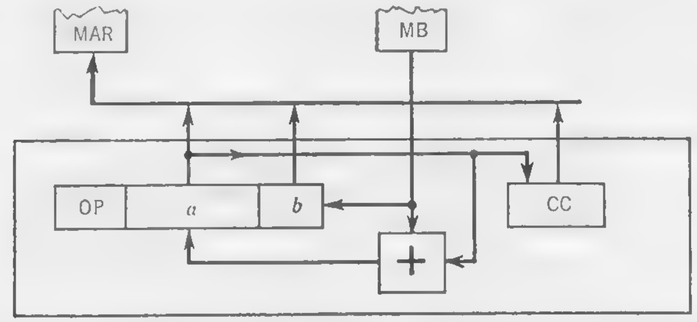


FIG. 16.13. Control for a single-address system with index registers.

The index part of the command could consist of four digits with the result that any memory location might play the role of an index register. In practice, however, *b* is usually not more than one digit and it is difficult to construct a sufficiently complex problem in which more than two digits would be needed for *b* (this gives nearly 100 index registers). When *b* consists of fewer than four digits the transfer $b \rightarrow (MAR)$ should set the other positions of MAR to zeros, say.

In the simplest form a special register is used for the index instead of part of the main memory. This may lead to a somewhat faster system since one less memory access is required but there is also less flexibility.

0. (CC) → (MAR).
1. ((MAR)) → (MB) → (CR)
2. b → (MAR), (b) → (MB), a + (MB) → a,
3. What is OP?
   (*a*) AD: a → (MAR), (a) → (MB), Add, to 4.
   · · ·
   (*j*) TN: { (Acc) < 0: a → (CC), to 0.
   (ACC) ≥ 0: to 4.
1. (CC) + "1" → (CC), to 0.

FIG. 16.14. Single-address system with index registers.

Index registers such as that described above make it easy to write programs that deal with subscripted variables such as a_i or $a_{i,j}$ (sometimes written $a[i]$ or $a[i,j]$) or with elements of vectors or matrices.

16.12. ELEMENTARY LOGICAL CIRCUITS

The above sections have discussed the general structure of digital-computing systems. Before details of such systems are discussed it is necessary to present some basic circuits in logical form which might be used in fabricating such systems.

Figure 16.15 shows examples of a number of elementary digital circuits. The first of these (Fig. 16.15*a*) is a flip-flop which stores a single binary bit of information. If the flip-flop stores a binary 1, the terminal labeled A is high and the terminal labeled nA is low in potential. After a signal has appeared on A_s, the terminal A will be high and the flip-flop will be said to have been set. A signal on A_r resets the flip-flop and after a reset signal nA is high and A is low.

The circuit in Fig. 16.15*b* is called a logical AND circuit. If both a and b are high then c is high. In terms of binary digits, if a and b are equal to 1, c is equal to 1; otherwise c is equal to 0. Therefore, c can be expressed by the formula $c = a \cdot b$.

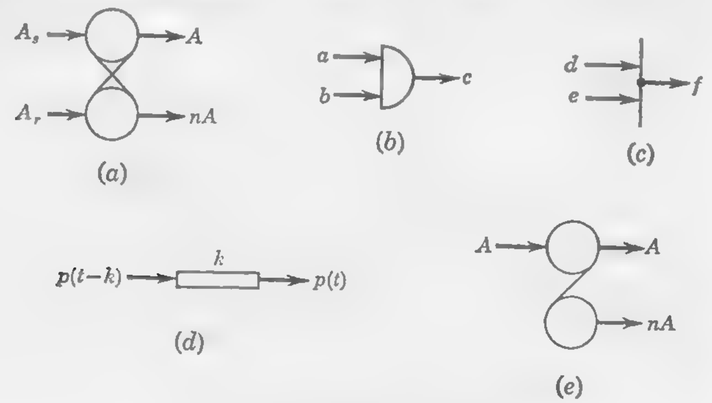


FIG. 16.15. Digital circuits.

The circuit of Fig. 16.15*c* is called an OR. If either d or e is high, then f is high, or the effect can be expressed by the formula $f = d + e$, if it is agreed that $1 + 1 = 1$. This type of algebra is called Boolean algebra.

The circuit of Fig. 16.15*d* is a delay circuit. If a signal $p(t - k)$, that is, a pulse train at time $t - k$, is applied to the input and the delay is equal to k clock-pulse intervals, then the signal appearing at the output would be $p(t)$. Thus, if a pulse is put into the delay line at a given time, it appears at the output k units of time later. This description has been in terms of pulses; such a circuit works equally well with respect to d-c signals, or gating signals, which last for more than a single clock-pulse time.

The circuit of Fig. 16.15*e* is called an inverter and has the effect of developing the complement of a given signal. For example, if A is high, then nA is low, or if A is a 1, nA is a 0. Likewise, if A is a 0, then nA is a 1.

16.13. A MEMORY UNIT

By making use of the logical elements as described above, and as shown in Fig. 16.15, it is possible to show a device which will store a number of pulses. This is a recirculating memory and is shown in Fig. 16.16.

In Fig. 16.16 information may appear on the input line and if the input control is high this information feeds through the AND gate and through the OR gate into the

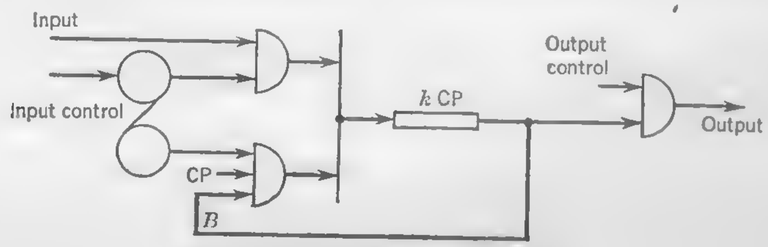


FIG. 16.16. Recirculating memory.

delay line. After coming out of the delay line it comes back to an input gate which allows it to enter the line, provided the input-control signal has now been removed. By supplying a proper input-control signal it is possible to transfer an arbitrary pulse train of appropriate length into the delay line. Likewise, an output-control signal can cause the pulse train stored in the delay line to appear on the output terminal. This type of recirculating memory system characterizes the type of memory used in the early delay-line computers, such as the SEAC, which was built at the National Bureau of Standards.

It is possible to transform the memory circuit of Fig. 16.16 into a serial accumulator, as shown in Fig. 16.17, by inserting a binary-addition circuit in the recirculation loop. The effect of the binary-addition circuit is to take pulse trains occurring on two inputs and to put out the pulse train, which represents the binary sum of the numbers represented by the pulse trains on the input. Reference to Fig. 16.17 shows an ADD and a

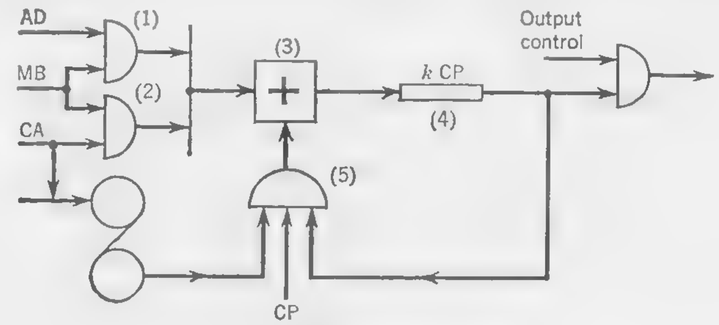


FIG. 16.17. Serial accumulator.

CLEAR AND ADD input to the serial accumulator. There is also an input labeled MB, which corresponds to the memory buffer shown in Fig. 16.5. If an ADD signal is present, then information from the memory buffer (which might be a serial memory unit such as shown in Fig. 16.16) passes gate 1, enters the adder 3, and the sum enters the delay line 4. If there is a CLEAR AND ADD signal, the information from the memory buffer passes gate 2 and the inverter inhibits gate 5 so that new information is fed into the delay line.

16.14. PARALLEL AND SERIAL REPRESENTATION OF INFORMATION

In a computer, a word might be stored in an array of 36 flip-flops (Fig. 16.15*a*), each flip-flop being set to either its 0 or 1 state in a one-to-one correspondence with the information shown. When represented in this manner by a spatial array of storage elements, the information is said to be stored or represented in *parallel*, as all digits of a given word may be inspected simultaneously.

Alternately, the whole word may be stored in a memory unit such as discussed in Sec. 16.13. In this case, the point B (Fig. 16.16) first assumes the state corresponding to the least significant digit of the word and remains in that state for a short predetermined time interval, then assumes the state of the next most significant digit, and so on until it has sequentially represented each digit of the whole word. Information represented or stored in this manner is said to be stored serially, as only one digit is available at a time. A clock (CP) is again commonly used to govern the rate at which digits of the word are passed through or shifted into the circuit. In the case of a dynamic flip-flop, this clock is usually the same as that used to control the flip-flop itself.

A "word" in a computer may thus be represented either in parallel as a spatial arrangement of a simultaneous pulse pattern (or voltage levels), or it may be represented serially in the time domain on a single pair of terminals as a sequence of pulses (or voltage levels).

Various combinations of the two methods mentioned above are also possible. In

this case, the information within the computer is said to be represented in a series-parallel fashion. It is, for instance, common that the binary digits comprising a character are represented in parallel, while the characters are represented serially within the word structure.

16.15. PERIODICITY OF INFORMATION FLOW

In any case, whether represented serially or in series-parallel, the word structure, being fixed, imposes a certain periodicity onto the information flow. To facilitate processing and transfer of information, these periodicities may be marked through the use of additional timing sources, denoting, for instance, character rates or word rates, or both. These additional timing sources are always submultiples of the master clock, and generally synchronous and in phase with it. The use of a "word clock" to control information flow within a given section of a computer is common and leads to efficient information exchange within that section.

16.16. TRANSLATION OF INFORMATION FROM SERIAL TO PARALLEL REPRESENTATION AND VICE VERSA

Parallel representation, on the other hand, imposes no periodicity related to the structure of the word, and machines organized in this fashion are often asynchronous in their operation. The choice as to whether information is organized into serial or parallel information patterns depends both on the means used to represent the information and on the logical function to be performed by that specific portion of the machine. Thus, in one section of a computer one may find the information arranged

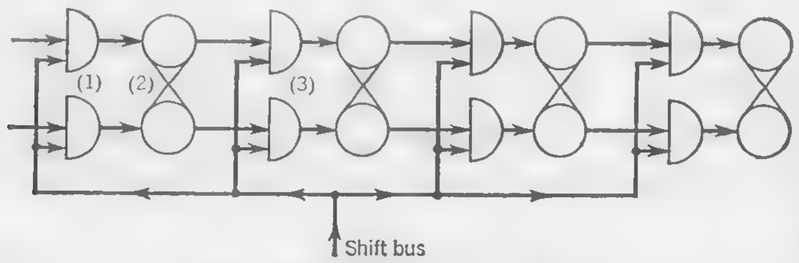


FIG. 16.18. A flip-flop shift register.

serially, while in other areas, a parallel representation is called for. Translation from one type of representation to another is accomplished in buffer memories, often of the shifting-register type. Typically, such registers consist of a series of flip-flops which can accept and deliver information in parallel as well as being capable of shifting the stored information pattern, spatially, either to the left or right, one digit space at a time. A flip-flop shifting register of this type is shown in Fig. 16.18.

16.17. TIMING AND SYNCHRONIZATION OF MEMORIES

16.17.1. General. The rate at which specific information can be transferred to and delivered by a memory is often the factor limiting the computational speed of a system. Careful organization of the memory for effective information transfer to other portions of the computer is essential.

16.17.2. Influence of Physical Characteristics of Memories on Timing and Synchronization. The memories in practical use today can be categorized as either permanent or transient, depending on their nature.

Memories based on physical properties such as wave-propagation phenomena (acoustic, electric, etc.) and charge-decay phenomena fall into the transient class. Memories based on polarization properties (magnetic, electric), electromechanical

devices, etc., fall into the permanent class. The table below shows a classification of the memories now most commonly used.

Permanent Memories	*Transient Memories*
1. Magnetic-core memories of parallel coincident-current type	1. Delay-line memories *a.* Acoustic *b.* Magnetostrictive *c.* Electric
2. Magnetic-core shifting register	2. Charge-storage-tube memories
3. Magnetic drums—parallel, serial	3. Diode-condenser memories

16.17.3. Periodicity Imposed through Regeneration. Transient memories require regeneration. That is, before the information decays or gets distorted to an unrecognizable form, it must be read and rewritten so that the original pattern (wave, charge, etc.) is reestablished. This regeneration process is normally carried out systematically, bit by bit, or word by word, and thus imposes a certain cyclic character upon the memory.

Memories of the transient class are therefore usually organized to operate in a synchronous fashion. This is particularly true of memories based on wave-propagation phenomena, as this characteristic also imposes stringent periodicity requirements on the information patterns.

Memories of the permanent class, such as those based on magnetic or electric polarization, require no regeneration. These memories can be organized in either periodic or aperiodic fashion and made to operate either synchronously or asynchronously with other sections of the computer.

16.17.4. Periodicity Imposed through Scanning and Selection. As mentioned before, the memory must be able to deliver or accept information to or from any arbitrary storage cell called for. Rather than endowing each storage cell with the capability of producing an electric output signal, many memories are so organized that reading and writing can be carried out only at a few specific stations. Access to the various memory cells is had here by periodically having the reading and writing stations scan the entire memory in sequential fashion. Reference to a specific cell may be had only periodically.

Thus it is evident that the selection mechanization may also impose a cyclic character upon the memory, independent of whether the memory is of transient or permanent nature. Magnetic-drum memories which belong in the permanent class usually are organized into a synchronously operating memory system in this manner.

16.17.5. Access Time and Information Transfer. The retrieval of information from a memory system involves at least two, and sometimes three, distinct processes, each requiring a characteristic time interval for its completion.

The first consists of the time required to get access to the specific set of memory cells from which one wants information. This time interval is often loosely termed "access time" T_A.

The second process involves the read-out of the information once the proper memory cells have been located. The time required for this process may be called the characteristic interrogation time T_I of the memory.

Information read-out of some memories involves a destructive process. Therefore, if the information is to be retained in the memory, it must be rewritten. The time required for this process has been called "latency time" T_L expressing the thought that for this period the memory is latent and no new information may be requested of it until this process is completed. As far as economically possible, the designers of computing systems strive to keep the three characteristic times mentioned as short as possible.

The physical characteristics of a memory system have been shown to impose definite patterns on the information flow to and from a memory. T_A, T_I, and T_L are usually closely related to the physical structure of the memory, although the information organization also influences these parameters in a significant manner.

Thus, for a memory based on wave-propagation phenomena, T_A depends on the

length of the delay line used, whereas T_I depends on the frequency response of this delay line, i.e., on how narrow pulses may be propagated down the line. T_L for circulating memories of this type is zero.

For memories of the magnetic-drum type, T_A depends on the rotational speed of the drum and T_I on the time response of the transducers, i.e., how fast a binary digit can be recorded or read. T_L is again zero, as the information stored is permanent and not destroyed upon read-out. In the cases cited above, T_A is usually many times (often hundreds of times) greater than T_I. An increase in information communication rate with the memory is had only through providing as many parallel communication channels as economically feasible under the specific situation at hand.

Memories of the permanent class, such as ferroelectrics and memory cores, usually are organized such that T_A is of the same order of magnitude or even less than T_I. This is possible when each memory cell has the capability of producing an electric output signal and the selection of a specific cell involves only electronic switching of terminal leads. Thus, in a typical core memory, T_A depends on the speed of the electronic switching circuits, but T_I depends on the inherent properties of the core material.

Information read-out of magnetic cores is usually destructive. T_L depends primarily on the characteristics of the core material and is usually close to T_I in magnitude.

A table of characteristic times for the more common memories is given below:

Memory	Typical values, μsec		
	$T_{A_{max}}$	T_I	T_L
Magnetic drums	500–50,000	4–50	0
Acoustic delay lines	50–500	0.5–5	0
Electric delay lines	1–50	0.1–2	0
Magnetic cores	1–5	1–5	1–5
Charge-storage-tubes	1–4	0.1–1	2–5

The value shown for T_A is the maximum value encountered. Depending on the particular instant information is requested from the memory, the access time T_A may vary all the way from zero to the value indicated. For a random request rate, the average value of T_A would be $\frac{1}{2}T_{A_{max}}$.

16.18. TRANSFER OF INFORMATION BETWEEN MEMORIES OF DIFFERENT CAPABILITIES

A memory embracing all the desirable properties needed for computer applications has not yet been developed. Thus the associated complexity of high-speed memories demands that their capacity be limited. High capacity at low cost and less complexity (for each binary digit) as yet can be had only in medium- or slow-speed memory devices. Several compromises have been worked out for those cases where both high speed and large memory capacity are required. These in general involve the use of more than one type of memory in the computing system. A high-speed low-capacity memory may be used to communicate directly with the arithmetic unit, with an auxiliary low-speed memory replenishing data in the high-speed memory as it is used up. The slow memory may either communicate directly with the high-speed memory or communicate with the high-speed memory via the arithmetic unit. The mode of operation is commonly such that block transfer of information from one to the other is involved. As the significant time parameter characterizing the low-speed memory is usually the access time T_A, the simultaneous transfer of a large number of words

results in a timewise efficient process. This is so because, although T_A may be large, the average digit time T_I from memories in this category still is small, and once the information to be transferred has been located, a high transfer rate can be maintained.

In achieving efficient information transfer to and from the two types of memories, one again is faced with the problem of synchronous vs. asynchronous operation. Here again, both schemes are currently in use. If the two memories chosen are of such character that they demand their own internal timing clock, only asynchronous operation is possible. Such is generally the case, for instance, in the operation of magnetic drums and acoustic delay-line memories. If, on the other hand, only one of the memories demands operation on its own internal clock, then this same clock, or a multiple or a submultiple of this clock, may be used to govern the operation of the second memory. In this case, synchronous operation between the two memories can be achieved. Examples of this type of operation include a magnetic-drum memory operating together with a magnetic-core or a cathode-ray-tube storage system.

Asynchronous operation demands the use of some buffer storage. This requirement may vary all the way from a single binary digit to a storage for complete words and even blocks of words. Needless to say, if the requirement for buffer storage goes up, the complexity of the associated equipment may increase to a point where it no longer is economically feasible to operate the two memory systems together.

A second need for buffer storage between two memories arises when the organization of the information in the two systems is different, say serial in one case and parallel in the other. Thus, for instance, information storage in a magnetic-drum memory is conveniently arranged in serial fashion. On the other hand, information in the high-speed memory often is arranged in a parallel fashion to achieve speed. Here, a buffer register of at least one word is required between the two memories so that the information may be translated from a serial to a parallel pattern; the shift register of Fig. 16.18 can be used for this purpose. As the arithmetic unit often is idle during this process of information transfer between memories, the storage registers of the arithmetic unit may be used for this buffering purpose. Most of the high-speed high-capacity computers on the market today have found it necessary to rely on more than one type of internal memory.

16.19. TIMING AND SYNCHRONIZATION OF ARITHMETIC ELEMENTS (GENERAL)

The arithmetic element in most general-purpose and special-purpose digital computers consists of several storage registers and a logical unit called an adder (or subtracter).

The timing problems in the arithmetic unit of a machine can be divided into two basic classes. The first is the control of the transfer of the information when it is being shifted from one arithmetic register to another or when it is flowing between the arithmetic unit and storage or input-output equipment. The second timing problem is associated with the functioning of the adder (or subtracter) and has to do with the timing related to the carry (or borrow) operation. The second problem is usually important only in parallel arithmetic units where a long carry through many stages may occur.

Synchronization and timing problems are significantly affected by the choice of equipment used to construct the storage units that make up the arithmetic registers. If a static binary storage element is used (e.g., a flip-flop or magnetic core) and, if two static storage units are associated with each binary position in the register, then synchronization and control-pulse-duration problems can be virtually eliminated. The machine can be essentially asynchronous. That is, control pulses can occur nonperiodically and be arbitrarily long. Any single storage unit must not, of course, be given simultaneous conflicting commands.

Timing and synchronization become somewhat more critical if dynamic, continuously regenerated, binary storage devices are used. Then information transfer can occur only at definite times. The timing is usually under the control of a constant-frequency control-pulse generator called a clock. Such a machine is synchronous.

The use of delay devices (electrical, acoustic, or magnetic) as the storage units of the arithmetic registers is common in machines that operate serially. Pulses representing individual digits will arrive at a given point in the system in sequence. Precise periodic synchronization of operations is essential.

As a general rule, a machine may be made faster and perhaps more reliable if asynchronous operation is used even though more equipment is involved. A machine that handles information serially and is synchronous will, in general, be slower and less expensive and will use less equipment.

The number of components in a machine can be significantly reduced by using a portion of the main storage unit for the arithmetic registers. An outstanding example is the use of "circulating registers" on a magnetic drum.

16.20. SERIAL ARITHMETIC-ELEMENT TIMING AND SYNCHRONIZATION

In transferring information in a serial computer, time is a necessary coordinate in the identification of the individual bits of each word, since the successive bits are sent

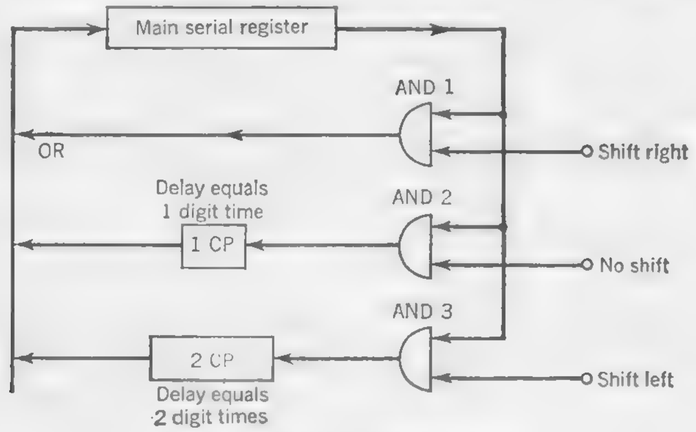


FIG. 16.19. A serial shift register.

serially over a single path. Note that time is also a coordinate defining information in a parallel machine, but an entire word is identified with a particular time and the separate bits of a word are transferred in parallel over many paths. A serial arithmetic unit has the advantage that the adder need handle only one pair of operand bits (plus a possible carry) at a time. Carry delay presents only a very minor problem in a serial arithmetic unit since the logical operation of forming the single binary digit representing the sum and the single binary digit representing the carry is simple enough to allow it to be performed very rapidly.

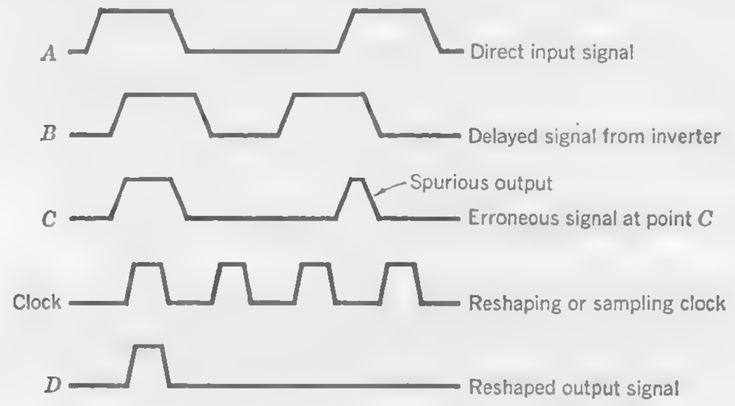
Most serial machines use a constant-frequency clock for the control of internal operations. This is not fundamentally necessary. Static storage can be used thus allowing aperiodic timing of basic operations. Usually the various basic transfer operations require approximately the same amount of time and there is ordinarily no particular advantage to be gained by using an aperiodic source of control pulses. Communications with input-output equipment of on-line (real-time) computations may sometimes be made easier by adopting aperiodic timing.

Shifting in a serial machine is most commonly accomplished by the insertion or deletion of "external" storage units in the serial path. Assume that information is being circulated from left to right through a serial register of Fig. 16.19. The output emerging from the right-hand end is fed back through an external path to the input. No shift may be defined as one unit of storage allowed in the external path. (It is assumed in this definition that the shift register is $N - 1$ positions or clock-pulse periods in length and that a "minor cycle" always consists of N clock pulses for one word time.) If two units of storage are included, the word in the register will be shifted one place to the left with each circulation. If the external unit has no storage

then the word will be shifted one place to the right, with each minor cycle consisting of N pulses.

Many serial machines use delay-line storage devices both for main storage and also for the arithmetic registers and are therefore highly synchronous in their operations. The "external" units of storage in Fig. 16.19 will ordinarily be static storage (such as a flip-flop) to simplify synchronization and timing.

Much of the hardware in data-handling systems is concerned with making sure that the output of a given gate is presented only when the various inputs to the gate are



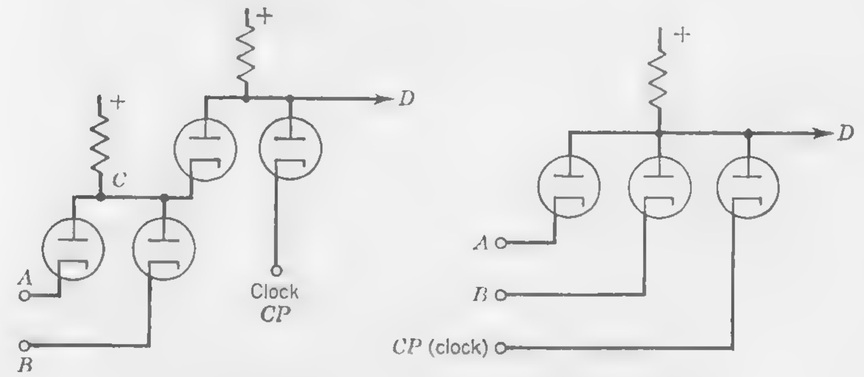


FIG. 16.20. Signals in a two-input clocked AND gate.

guaranteed to be at meaningful levels. The determination of the exact time to sample the combination of inputs becomes more difficult as one presses for higher operating speeds.

The logical design of any machine, serial or parallel, will require the combination of several signals as inputs to one AND gate. The signals may originate in widely separated parts of the machine. Further, the logical negation or inverse operation is frequently required. A slight delay ordinarily occurs while the inverter operates when this is realized with a tube or transistor device. The arrival of the delayed, inverted pulse at the input to a gate may not be synchronous with other pulses which have not been inverted and therefore not delayed. The output of the AND gate may be made narrower than the input, or a narrow pulse may sneak through when none should. Synchronous systems therefore require that a given logical output function of several input signals be reshaped or "reclocked" before it can be used. The logical function is strobed (sampled, multiplied) by a narrow clock pulse. This reshaping clock pulse is timed to occur during the period when the output signal is guaranteed to have the correct value. The resynchronized output pulse, D of Fig. 16.20, is necessarily narrower than the input pulses A and B which are sampled. If the process

of producing an output narrower than the input is repeated through several stages of gates, with the sampling pulse being made narrower each time, the information will soon disappear altogether. Therefore, before an output signal, such as that represented by D in Fig. 16.20, can be used, it must be "stretched" back to the standard length represented by inputs A and B.

If it is important to operate at the highest possible speed, the basic period of a clock cycle can be subdivided into "phases." For example, a given vacuum-tube pulse reshaper may take ¼ μsec to operate. If the basic clock period is 1 μsec, then ¾ μsec would be wasted by waiting a full period to reclock the output. A clock pulse of the proper phase (delayed ¼ μsec from that used to form the inputs to the reshaper) is therefore used to reclock the output.

This basic problem is present in all synchronous machines—serial or parallel. It has perhaps been magnified in serial-machine design by the desire to operate at the fastest possible basic information-transfer rate in order to make up for the inherently slower over-all information-processing speed of the serial machine.

16.21. PARALLEL ARITHMETIC-ELEMENT TIMING AND SYNCHRONIZATION

In a "parallel" arithmetic element, all the binary digits of a word are manipulated simultaneously. A single command pulse from the control unit is all that is necessary to accomplish the transfer of an entire word of, say, 36 bits from one register to another. To allow this, it is necessary to provide 36 separate paths through 36 gates. The control leads of all the gates are tied together to receive the command. Examples are given in Secs. 16.22 through 16.25.

Shifting and transfer operations in a parallel arithmetic unit can be self-timing or asynchronous. Both the duration of and period between command pulses can vary over wide ranges. Synchronization and sequencing of operations in this type of system can have a self-checking or interlocking property. That is, the control may be arranged to assure that the $(n + 1)$th step cannot start until the nth step is successfully completed. When this is done, precise matching or compensating for transmission delays or operating times is not necessary.

Some large parallel machines have been built in which the *only* synchronous arithmetic-unit operation, in the sense that it always takes the same amount of time and is not determined logically, is that associated with "carry delay." This is the delay of a subsequent operation for a time sufficient to allow for the longest possible carry.

Many parallel machines use a synchronizing clock for timing basic internal operations. In this case the control unit can supply command pulses to the arithmetic unit only at predetermined periodic times. The clock-pulse period is ordinarily chosen to be the least time that can be safely allowed between the elementary arithmetic operations such as clearing a register or gating or transferring information from one register to another. An operation that requires more time to execute is handled by inhibiting the issuing of the next command signal for the necessary number of clock-pulse periods.

Timing and synchronizing methods of arithmetic elements also differ on a higher level. A machine can use a periodic clock as a source of basic timing signals but use different numbers of periodic clock cycles to execute complete arithmetic operations depending on the value of the operands. The operation of the machine can be synchronous in the timing of the basic or elementary steps but still be asynchronous in the over-all operation of the arithmetic and control sections.

For purposes of illustration, consider the way binary multiplication is commonly performed. If the current multiplier digit is a 0, the partial product in the accumulator is merely shifted. If the multiplier digit is a 1, the multiplicand must be added to the partial product in the accumulator, time must be allowed for carry to propagate, and finally the new sum in the accumulator is shifted. These two cases will not take the same amount of time, and the speed of the machine can be increased if advantage can be taken of this fact.

Many machines are entirely synchronous. The execution of each type of arithmetic operation always takes a fixed length of time, independent of the value of the operands. The advantage of this arrangement is the relative simplicity of the control unit. Also, communication with a synchronous storage or memory unit is much easier to arrange.

16.22. TRANSFER AND SHIFTING IN DOUBLE REGISTERS (STATIC PRIMARY STORAGE)

In the series of machines constructed according to a basic philosophy developed at the Institute for Advanced Study, information is never stored in the transient state. This premise leads to the "double-register" design (Fig. 16.21). If a number in

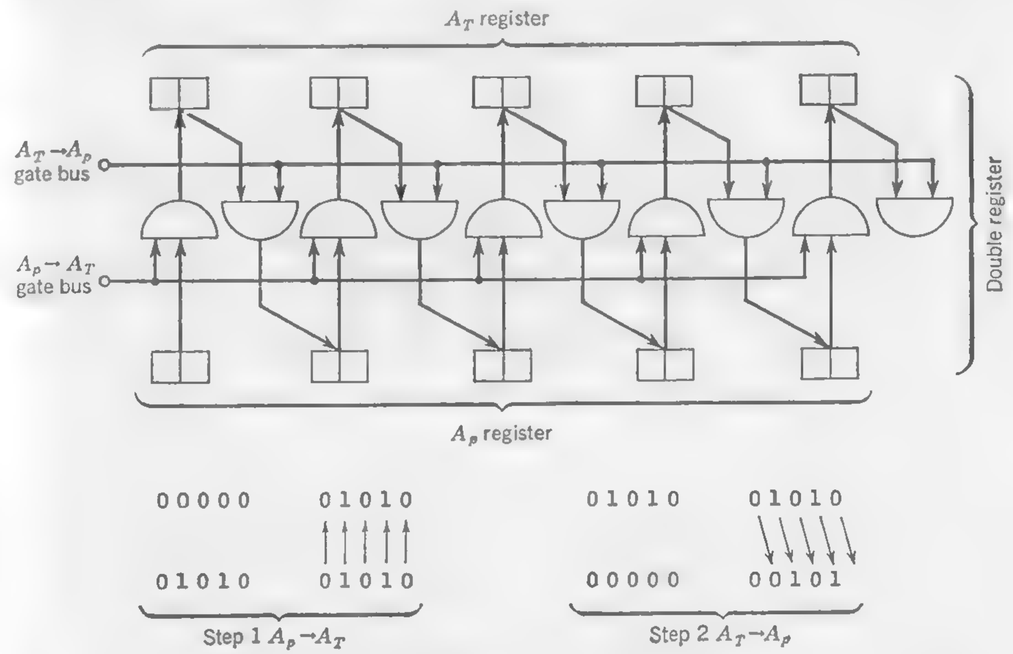


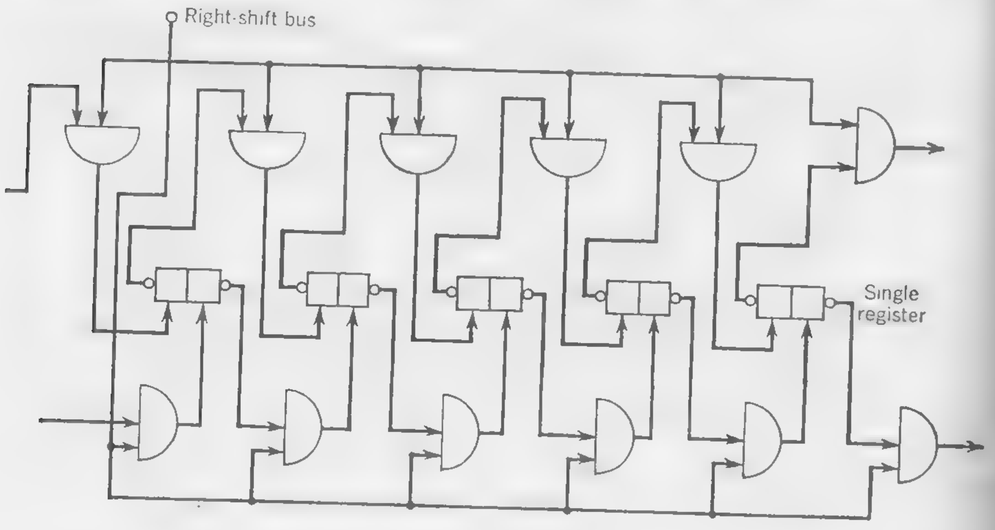
FIG. 16.21. Shifting in a double register.

register A_p is to be shifted one place to the right, this operation takes place in basically two steps. The number is first transferred or gated directly (without shifting to the left or right) into the temporary register A_t (clearing out any information that may have been in A_t). In the second step, the information is transferred back into A_p, with the appropriate shifting taking place. The old information in A_p is cleared out in the process. The important things to notice are: (1) the gates connecting A_p to A_t may be open for an arbitrarily long time in step one; (2) the shifting gates connecting A_t to A_p may be open as long as you please; (3) the second gate signal can start immediately after the termination of the preceding gate signal. The gating signals must be long enough to allow the receiving register to be properly set but there is no harm if they are longer. The removal of the upper bound on control-command pulse length makes the control timing problem less critical. It is also possible to use error-detecting interlocks. Various methods for generating and controlling the timing of gating pulses for double-register operation are described in Sec. 16.32.

The storage elements in the double-register scheme need not be constructed of tube or transistor bistable circuits. The static magnetic delay line as originally conceived used two magnetic cores per bit, one for temporary storage and one for permanent.

16.23. TRANSFER AND SHIFTING IN SINGLE REGISTERS (STATIC PRIMARY STORAGE)

The duplicate temporary storage register equipment required in the double-register scheme may be eliminated by using transient storage of information during the shifting process. A commonly used system is illustrated in Fig. 16.22. Assume that the flip-flop has been in the state with A_i high and B_i low for some time. Current at points A_i and A_o will both be high; B_i and B_o will both be low. If the state of the



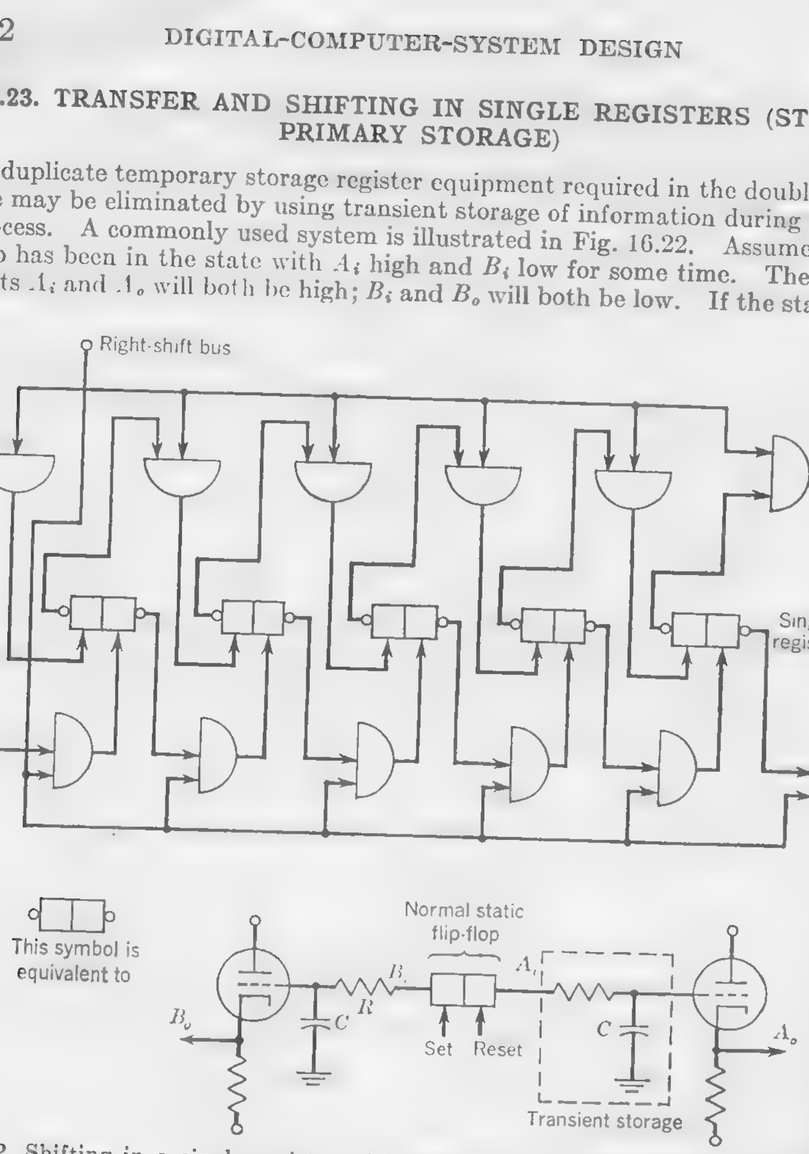


FIG. 16.22. Shifting in a single register with static primary storage but transient (RC) temporary storage.

flip-flop is changed (A_i low and B_i high) the previous voltage levels will be "remembered" for a short time in the capacitors such that A_o and B_o will remain substantially at their previous voltage levels. A gating signal may be used to connect the delayed output signals to the input of another flip-flop even though A_i and B_i are changing. This gating signal must terminate before the output signals being transiently remembered have had a chance to change appreciably. The next operation cannot proceed until a sufficient time has elapsed to allow the capacitors to charge to the new values such that the output signals (A_o and B_o) agree with the signals at the flip-flop (A_i and B_i).

Relatively inexpensive flip-flops may be caused to change state in a fraction of a microsecond. This means that the gates determining the new state of the flip-flop need to be sampled only for a fraction of a microsecond. If a charging time for the RC transient storage network is made to be several microseconds, the value of the inputs to the gates will not change appreciably during the short sampling (or clock-pulse) time.

Other means of temporary storage may be used. In particular, the (usually

undesirable) minority carrier-storage effect in transistors can serve as temporary information storage.

Machines with arithmetic registers based on transient storage of the type described above require a shifting control or "clock" pulse of a definite duration, but the repetition rate need not be fixed. There is a lower limit to the time required to charge the storage capacitor to the new value. There is, however, no upper limit to the length of time between successive "clock" pulses. In this sense the operation may be asynchronous. However, the sampling or clock pulses usually are derived from a constant-frequency oscillator as the prime source.

Another form of temporary storage is the delay line. Assume for definiteness that each flip-flop has associated with it a 1-μsec delay line (Fig. 16.23). At time T_o, the

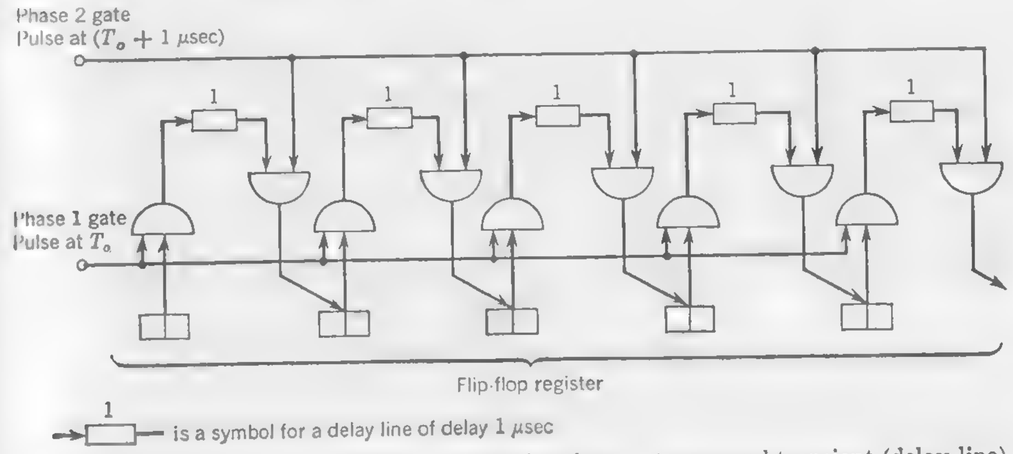


FIG. 16.23. Shifting in a single register with static primary storage and transient (delay-line) temporary storage.

phase 1 gates are enabled, and all digit positions with flip-flops containing 1's will allow pulses to enter the corresponding delay lines. The phase 2 gates are then operated at T_o + 1 μsec just as the information comes out of the delay lines. Any 1's are gated back into the flip-flop register but shifted one place to the right. It is assumed in this example that the operation of reading from the flip-flop register into the delay lines resets all flip-flops to zero. Both the duration of the two gate enable pulses and the separation in time (1 μsec in this example) must be accurately controlled.

If the phase 2 gate signal lasts too long (for instance, longer than 1 μsec) the information in the delay lines will no longer be present.

16.24. TRANSFER AND SHIFTING (DYNAMIC OR REGENERATED PRIMARY STORAGE)

The 1-μsec delay line in the preceding section which was used for temporary storage can be used as primary storage. The flip-flop may be eliminated. This arrangement is referred to as a "dynamic toggle" or dynamic flip-flop.

In the machines patterned after the SEAC, the delay is a true distributed electromagnetic delay line. It is also possible for an exceedingly simple lumped delay line to be used. The IBM "microsecond delay unit" is perhaps the simplest possible form. Information is alternately stored in the magnetic field of a single inductance or in the electric field of a single capacitor. In either realization of a dynamic toggle, information is continuously passed back and forth or regenerated. In the delay-line version, resynchronizing clock pulses must be accurately periodic and the period must coincide with the delay of all delay lines used. In the microsecond delay unit, the timing of the transfer of information is relatively insensitive to the value of the

inductance and capacitor, and small variations of the clock-pulse period can be tolerated.

A dynamic-toggle shift register is logically similar to the double-register device described in Sec. 16.22. If information is being held in the register, gating is vertical between corresponding digit positions to afford the required continuous regeneration.

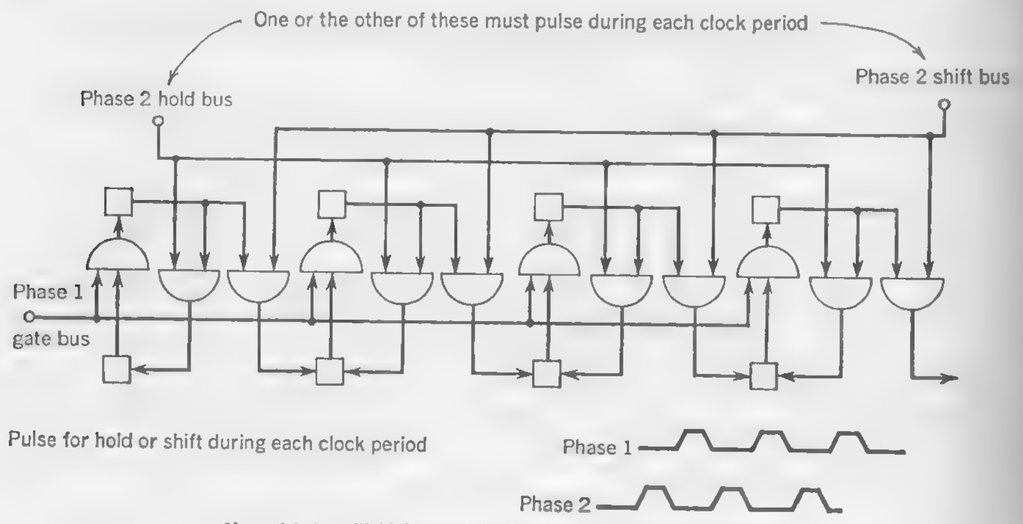

One or the other of these must pulse during each clock period
Phase 2 hold bus
Phase 2 shift bus
Phase 1
gate bus
Pulse for hold or shift during each clock period
Phase 1
Phase 2

FIG. 16.24. Shifting with dynamic primary storage.

If a shift to the right is to be made, a vertical transfer is followed by a diagonal transfer (see Fig. 16.24).

16.25. TRANSFER AND SHIFTING (DELAY-LINE PRIMARY STORAGE)

The use of a magnetic-drum delay line as primary storage for arithmetic registers is a popular way for producing a machine with a minimum number of active elements. The arithmetic registers in such a machine are essentially a special case of the main storage unit. The shifting and transfer of information between circulating registers themselves and between the circulating arithmetic registers and the main storage is synchronized by the inclusion of external static storage units which usually take the form of flip-flops.

Shifting of information is accomplished by including additional delay or removing delay in the external serial path of the circulating register, as was described in Sec. 16.20. These external storage units also provide a means for handling the problem of the relative delays between reading from a track on a magnetic drum and writing on the same track.

The one-word arithmetic registers may also be either an electrical or an acoustic delay line. The former has been used more often as an arithmetic register. The acoustic delay line has been used extensively for main storage. Synchronization of the delay times in arithmetic-register delay lines with those in the main storage unit can be a serious problem. It often requires the use of very accurate temperature control.

16.26. COMPARISON OF SHIFT-REGISTER METHODS—A-C VS. D-C COUPLING

Many machines use a-c coupling and transmit information as pulses. Usually the two conditions for a binary digit are: presence of a pulse (when one may be expected) means binary 1; absence of a pulse (when one may be expected) means binary 0.

There is a hazard peculiar to computing devices in using a-c coupling. The duty

cycle is often quite variable. It is a painstaking task to consider all possible combinations of duty cycles in a complex system in order to be sure that transformers and other energy-storage components used in a-c coupled circuits have had a chance to recover adequately in time for the next pulse. The advantage of being able to transmit a pulse of very short duration at low impedance over relatively long distances is very often offset by the need to wait for a settling period before another pulse can be sent. Some systems have attempted to solve this problem by always transmitting symmetric signals with no d-c component.

A-C coupling has the strong advantage that pulse transformers may be used. This allows good matching to low-impedance transmission lines. When a computing system becomes physically large, it is necessary to use transmission-line techniques and not depend on "chassis ground" or "frame ground" as a signal reference. A transformer can convert an ungrounded input from a distant source to a locally grounded signal to drive ground reference circuits.

When vacuum tubes are used, a minimum 1,0 signal amplitude of 5 to 10 volts is required to assure complete on-off transition. The signal is usually referred to a common "frame" ground and the amplitude must be large to avoid confusion from "ground noise." The capacity that is unavoidably associated with a long signal lead must be charged from a driver if the machine is to be fast.

Transistors are essentially current-operated devices. The input-voltage swings need, in principle, be only 1 volt or so. Greater care must now be taken to avoid trouble from ground noise. Circuit capacity charging is lower both because the voltage swings may be smaller and because the capacities may be considerably reduced by the compact packaging possible with transistors.

The use of direct-coupled circuits has become very popular as a means for avoiding duty-cycle difficulties. One disadvantage of d-c coupling is the difference in d-c level of the input terminal and output terminal of a vacuum-tube voltage amplifier. The ordinary resistance-divider coupling networks will attenuate the signal. This is further aggravated by the necessity of allowing for the drift of coupling resistor values. Most irritating of all is the fact that the output tap of the resistor divider is usually at a relatively high impedance. A cathode follower or other amplifier must be included before any significant load may be driven.

With transistors it is possible to use direct coupling without resistor-dividing networks between stages to reestablish the d-c level. The output collector may be connected directly to the input base of successive grounded emitter stages. With germanium transistors, this implies that the ON transistor is saturated and that the 1,0 signal level is very small (about 0.3 volt). Transistor saturation will make the basic operating time longer because of the necessity of removing stored minority carriers from the base region. The alternating use of *n-p-n* and *p-n-p* transistors in successive stages allows interesting direct-coupled nonsaturating circuit possibilities for very fast computing applications.

The complexity of the double-register d-c coupled system is justified by its apostles on the ground that only the d-c level of a gating signal is important—the duration can be arbitrarily long. Further, the amplitude can be any value larger than the threshold value. Too large a gating signal cannot produce an error.

Often, a mixed strategy is used in that "logical signals" are d-c coupled but the strobing clock pulse is a-c coupled. Since the clock signal is periodic, the duty-cycle problem is relatively easy to solve.

Several methods commonly used to construct arithmetic unit registers are compared in Table 16.1 on the basis of restrictions on gating-pulse duration and period.

16.27. TIMING OF THE CARRY OPERATION

Timing of the carry operation is a problem only where the carry must propagate through many positions in parallel machines. In a "simultaneous" parallel adder, the two operands are presented on two sets of wires. Let us assume that the adder is binary and, for definiteness, 40 digits long. Further, assume that one of the operands consists of 40 1's and the other consists of 39 0's and a 1 in the least significant

place. In this situation, a carry is produced in the least significant place and passed to the next-to-least significant place. A new carry is then produced and passed to the third, etc. In this example, a carry must "propagate" through all 40 binary stages. Since a finite amount of time must elapse for the signal levels in the various stages of the adder to change, the sum digit in the most significant place cannot be correct until the carry digit which started in the least significant place has affected all the digits in sequence and finally reached the most significant position.

Table 16.1. Types of Arithmetic Register Systems

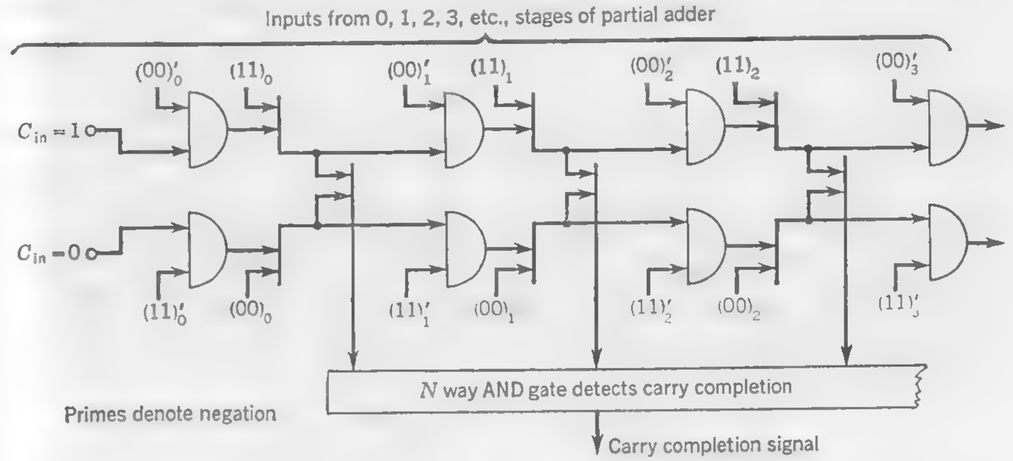
Type of register	Duration of gating pulses		Time between successive gating pulses	
	Lower bound	Upper bound	Lower bound	Upper bound
Double register with static storage elements	Long enough to set toggle	Unlimited	Immediate	Unlimited
Single register with *RC* temporary storage and static permanent storage	Long enough to set toggle	Must terminate as soon as possible	Wait for *RC* circuit to settle to new value	Unlimited
Single register with delay line (d_o μsec long) static permanent storage	Long enough to set toggle	Not longer than d_o μsec	Must occur d_o μsec after start of preceding pulse that gated information into the line	
Single register with dynamic toggle	Long enough to regenerate	Must terminate as soon as possible	An integral number of clock-pulse periods	
Delay-line register...	Long enough to regenerate	Must terminate as soon as possible	Set by delay time of line and number of bits held. Must be periodic	

One method for handling carry-propagation timing is to provide a "carry-delay" device which inhibits the initiation of the next step in the sequence of arithmetic commands long enough to allow the longest possible carry to propagate through the entire "simultaneous" adder. If the internal timing is controlled by periodic clock pulses, a sufficient number of clock pulses is allowed to tick by to guarantee complete settling of the adder network before a pulse is chosen to gate the output of the adder in the next step.

Since the speed of all four basic arithmetic operations depends on the basic carry time of the adder, it is worthwhile to consider methods for reducing this time. In the addition of two operands chosen with a random selection of digits, the number of places through which a carry (or borrow) must propagate is, on the average, only about 15 per cent of the full length of the adder. This fact allows one to increase the speed of operation by logical means after the choice of basic hardware (and therefore the fundamental transfer speed) has been made. Figure 16.25 is a logical diagram of one method for determining the occurrence of carry completion by logical means.[15]

Only the carry-determination components are shown; other elements form the combinations (00)′, (11)′, etc., of the two digits to be added in each position. Before the start of the addition operations, two special conditions must be met. First, the least significant carry input is not 1 and also not 0; i.e., the $C_{in} = 1$ and $C_{in} = 0$ lines are both OFF. This is a three-state device. The second condition is that the output of each digit position of the adder is forced to be in the (00)′ or the (11)′ state. For this case, all inputs to the carry-completion gate are OFF. At the start of an addition operation, the appropriate least-digit carry input is turned on at the same time as the operand (A_i,B_i) inputs are presented. An input to the carry-completion gate

will occur immediately for a given stage if the operand digits are both 0 or both 1 [(00) or (11)]. In either of these cases, the carry-output digit for that position is independent of the carry-input digit. The state of the carry for that stage is therefore immediately determined and the corresponding signal to the carry-completion gate is turned ON. If the local digits are not alike (00)′ or (11)′ then the carry-completion signal will be determined by a carry signal propagating from stages of less significance. Two carry-propagation lines are used—one for carry equals 1, the other for carry equals 0. As soon as a given stage receives a signal from the preceding stage, either that the carry digit is 0 or that the carry digit is a 1, then the local decision is made and an input for that stage to the carry-completion gate is turned on.



A_i	B_i		
0	0	$(00)'_i$	$(11)'_i$
0	1	$(00)'_i$	$(11)'_i$
1	0	$(00)'_i$	$(11)'_i$
1	1	$(00)'_i$	$(11)'_i$

A_i = The addend digit in the ith digit position
B_i = The augend digit in the ith digit position

FIG. 16.25. Logical scheme for detecting carry completion in a parallel arithmetic unit.

In calculators used for scientific computing or real-time control, multiplication is ordinarily the most common "long" operation. Division will almost always take longer but usually occurs less frequently. The information-processing speed of such a machine may be increased by rearranging the method by which multiplication is performed.

The over-and-over addition method of mechanizing multiplication can be speeded up by using an auxiliary carry register and an adder which does not propagate the carry all the way. This is useful in coded decimal machines. The carries generated in the Nth step (if addition without carry is used) can be added to the partial product along with the multiplicand on the $(N + 1)$th step. Each adder position receives two operand digits and a carry digit during each step. A separate one-binary-digit-per-decimal-digit carry register must be provided. If 10-decimal-digit multiplication is considered, the first 10 steps are accomplished using addition without full carry (thus avoiding the long carry delay). On the last step, the adder is connected in the "simultaneous" mode such that all carries are added in and accounted for simultaneously. This will speed up multiplication but is no help in division, where a full-length carry (borrow) must be used to find out if the trail division "goes" or does not "go."

16.28. TIMING AND SYNCHRONIZATION IN THE CONTROL SECTION—INTRODUCTION

In the familiar block diagram of a general-purpose digital computer, the control unit is indicated as a simple block. The arithmetic unit is another block. One may think of the arithmetic unit which processes the information as having a hopper for input information and chute for processed information and handles which control the processing of information. There may be one handle which, when operated, clears a register, resetting it to zero. Another handle may cause the contents of two registers to be added. A third may shift the number in a register one place to the right. It is the job of the equipment in the control block to manipulate the handles in the proper sequence in order to cause the steps necessary to process the information to be carried out.

A differentiation is needed between the concept of an "instruction" and that of a "command." An instruction is taken to mean a complete operation from the point of view of a programmer, or coder, for a machine. In a single-address parallel machine, the single instruction "reset and add absolute magnitude" might, for example, consist of the following seven commands: (1) Clear the accumulator register, resetting it to zero. (2) Transfer the word to be added (the addend) from storage into a register. (3) Examine the sign of the addend. (4) If the addend is negative, complement the contents of the register. (5) Add the number in the register into the cleared accumulator. (6) Signal the control unit that the execution of the instruction is complete. (7) Request the next instruction from the storage location determined by the program counter. More complicated instructions, such as multiply or divide, of course, will require many more commands.

16.29. SEQUENCING OF INSTRUCTIONS

The basic mode of operation of a machine is dependent upon the manner in which successive instructions are made available to the control unit. Various methods are discussed in detail in Secs. 16.6 to 16.11. Perhaps the simplest form of instruction sequencing is that used in wired-plugboard program-controlled machines. The nature of the successive instructions is determined by the locations chosen for the wires inserted in the plugboard.

Figure 16.26 represents a rudimentary arithmetic-control system for producing the command pulses necessary for the execution of the individual steps of an instruction. The control section must also generate a series of "program-step" pulses. These pulses are connected to the individual arithmetic-operation control units by wires inserted into the plugboard. The particular arithmetic-control operation chosen then routes the individual command pulses to the arithmetic unit and determines what each command pulse is to do. For example, "clear the accumulator," etc.

In Fig. 16.26, the command-pulse distributor and the instruction-pulse distributor can each be a special form of a shift register. In this device, a single 1 is transferred from position to position, sequentially. The end of the shift register is connected back to the input so that the 1 is circulated. This arrangement is also called a ring counter. Another method of distributing command-control pulses which requires less storage equipment but requires more logical or gating equipment is to use a binary counter, the output of which is decoded into the required number of output lines.

If we assume in Fig. 16.26 that the start-stop control toggle is set to the GO state, gate A_1 will allow successive clock pulses to cause the command-pulse distributor to step. With each step, a control pulse is emitted on the next line in the sequence. The device performs the function of a rotary switch.

For simple arithmetic operations such as addition and subtraction, each complete cycle of the command-pulse distributor can complete an instruction execution. For such a situation, the instruction-length counter will allow the last command-control pulse to pass through gate A_2 and cause the instruction-pulse distributor to step to the next position. In more complicated arithmetic instructions (like multiply and

divide) many cycles of the command-pulse distributor loop must be executed before the instruction is finished. Each cycle of the command-pulse distributor is referred to as a minor cycle.

In a stored-program-controlled machine, the instructions are not wired into a board but are stored as digital information in the main or auxiliary storage unit. The basic program steps are ordinarily carried out in two basic phases. The first involves the acquisition of the instruction from storage. This part is sometimes referred to as the "fetch." The second phase is called "execute."

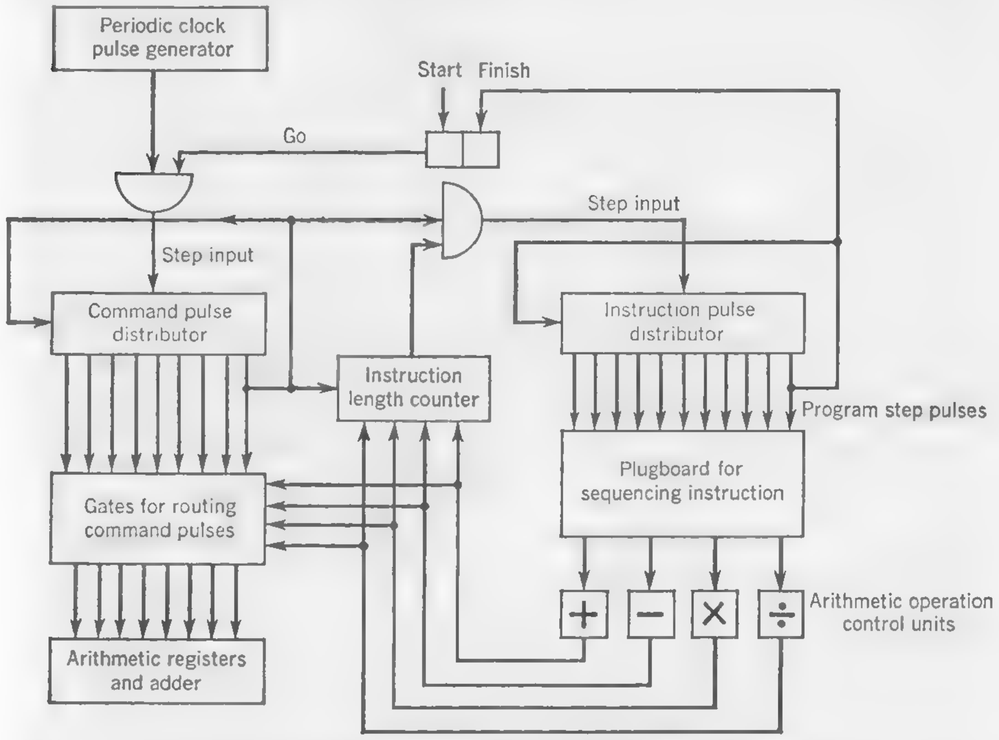


FIG. 16.26. Basic organization of a wired-plugboard computer.

The control unit must keep track of where it is in the sequence of instructions. The address, or main storage location, from which the next instruction is to be taken must be held in the control. In a single-address machine, instructions are ordinarily stored in sequential locations. A program counter is used to generate these addresses in sequence except where this normal flow is interrupted by a "jump" instruction. In the case of a jump instruction, the address for the next instruction to be fetched comes from the address part of the present instruction being executed (the jump instruction) instead of from the program counter. The state of the program counter must be changed to agree with this address so that it will count on from the new starting position determined by the jump instruction.

Part of the time allotted for the fetch operation is used to allow the meaning of the instruction to be interpreted by the operation decoder. The output signals from the operation decoder will prepare the state of gates in the control unit for the proper distribution of pulses during the second or "execute" part of the basic cycle (Fig. 16.27).

During the execute phase, command signals are routed to the arithmetic unit in accordance with the state of control gates established by the operation decode logic. Examples of particular realizations are given in the following sections.

The fundamental distinctive feature of a stored-program machine is the ability to alter its own instructions. This can be accomplished by having an instruction call

for information from a storage location which contains another instruction. This second instruction is read into the arithmetic register. The address part (for example) of the instruction may now be altered by adding a number from still another storage location to the address part of the instruction in the arithmetic unit. This altered instruction may now be returned to storage and used later on in the program as a new instruction.

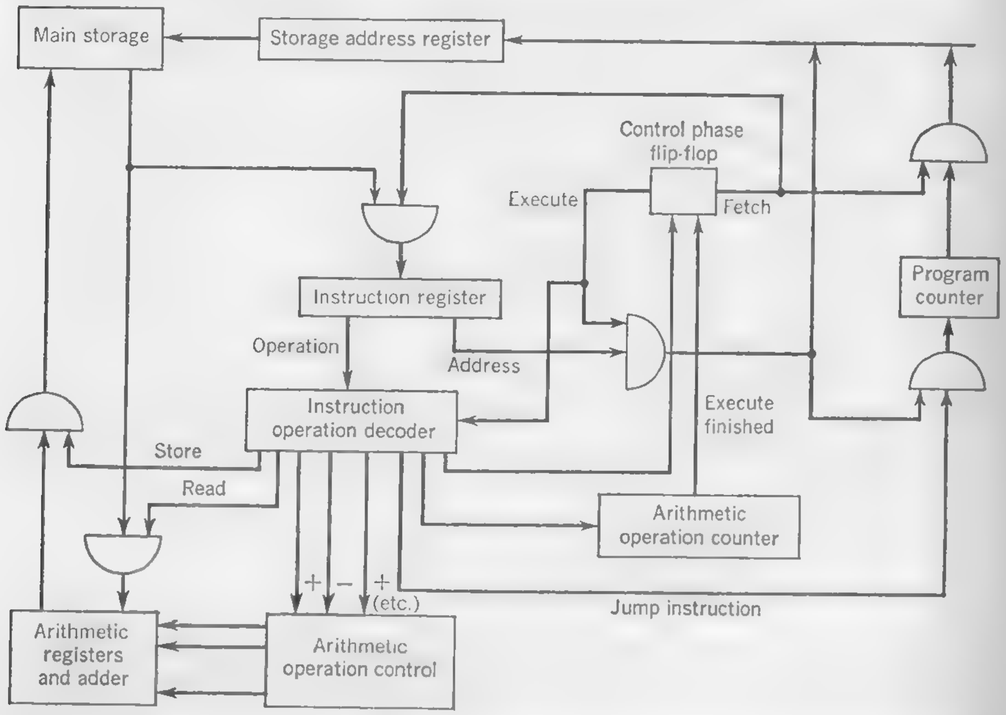


FIG. 16.27. Simplified diagram of control organization for a stored-program computer.

Some computers use the stored-program concept but do not allow instructions to be altered by the arithmetic unit. This is particularly true in digital computers designed for use in real-time control such as guidance of an aircraft. They are, in a sense, logically equivalent to the wired-plugboard-program machine.

16.30. SEQUENCING OF COMMANDS SYNCHRONOUSLY (FIXED MINOR CYCLE)

If high-speed operation is not the most important consideration, it is usually easier to have all simple instructions take the same amount of time. This basic unit of time is often called a minor cycle. Consider an elementary serial decimal machine where decimal digits are transferred serially by looking for a pulse during one of 10 pulse times. A minor cycle might consist of 25 clock-pulse times. This defines the length of the command-pulse distributor of Fig. 16.26. The first 10 are used for clearing registers (if necessary) and to set up gates according to the instruction to be executed. The next 10 clock-pulse times are used for transferring any numbers between storage registers and/or an accumulator. The balance of the clock-pulse times in the minor cycle are used for handling carries and other miscellaneous functions.

A fixed minor cycle is essential for serial or serial-parallel machines in which the basic storage element is a delay line or a magnetic-drum circulating register. Each minor cycle will control the transfer and identification of the successive digits of the words being manipulated.

Synchronous minor-cycle-control organization can also be useful in parallel computers. If a volatile storage unit is used (for example, Williams tube electrostatic storage) words may be transferred to or from storage only in synchronism with the basic regeneration timing. A minor cycle time equal to the basic regeneration period of the memory is convenient. Arithmetic operations are then constrained to occur in multiples of the basic minor-cycle time.

A fixed minor cycle may be a restriction in that some operations can be completed in less time. On the other hand, the design of a fixed-minor-cycle control unit is probably the least complex.

16.31. SEQUENCING OF COMMANDS SYNCHRONOUSLY (VARIABLE MINOR CYCLE)

The delays inherent in a fixed-minor-cycle system can be avoided by using only the number of periodic command pulses basically required for each operation. Consider a parallel machine with a minor cycle consisting of a maximum of eight pulses.

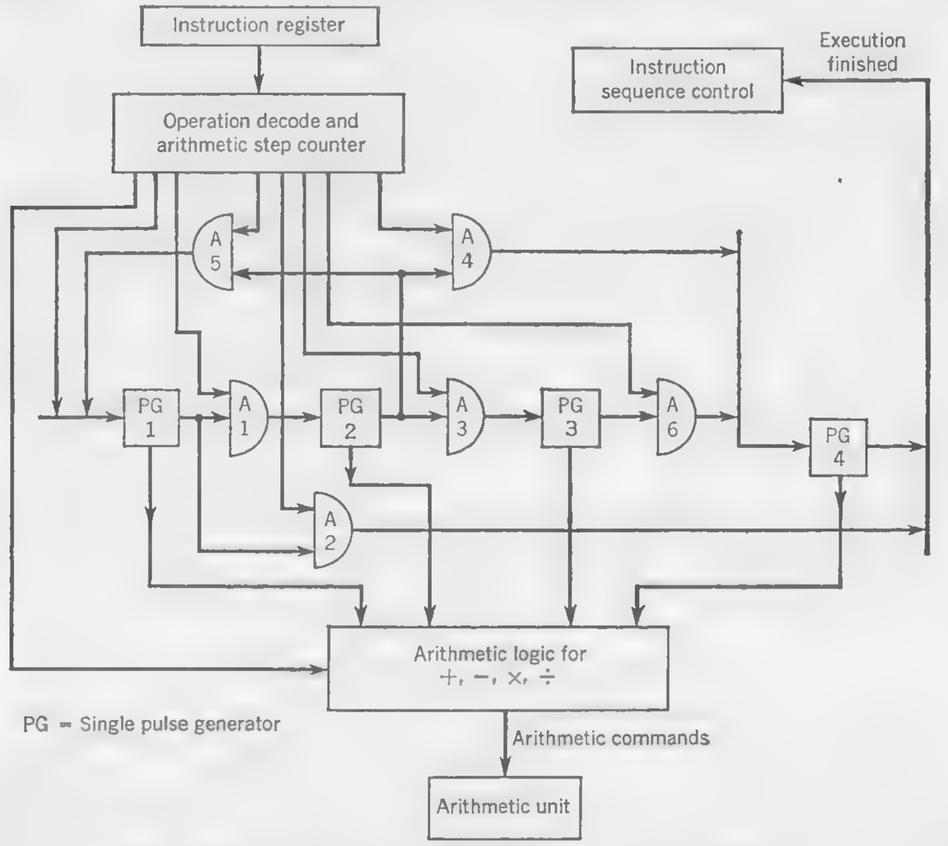


FIG. 16.28. Variable-minor-cycle command-pulse distributor.

If an operation is short (such as a jump instruction) and can be accomplished in five pulses, then, at the completion of the fifth pulse, a reset signal is sent to the minor-cycle pulse distributor. This same pulse is interpreted as a "completion-of-execution" signal by the major cycle or execute-fetch control.

Another variation has been used in which the duration of the individual command-signal pulses of a minor cycle may be different from each other. Such an arrangement is intended to make use of the possibility that clearing a register may take a different amount of time from that required to gate a number into a register. Successive command signals in such an arrangement are derived in the manner of Fig. 16.28 where each block can emit a command pulse of arbitrary length. The sequence in

which the pulses are emitted is under the control of the operation decoder. This arrangement of command-signal distribution is synchronous only in that the durations of the command-signal pulses are fixed and independent of the operation of the controlled circuits.

16.32. SEQUENCING OF COMMANDS ASYNCHRONOUSLY (SELF-TIMING)

It is possible to design a command-pulse distributor in such a way that it proceeds at a speed determined by the operation of the arithmetic circuits themselves. This arrangement has been used only with double-register machines (see Sec. 16.22). The basic shifting operation of such a machine may consist of four steps. These are clear the temporary register, transfer from permanent to temporary, clear the permanent register, and transfer (shifting) from temporary back to permanent. This fundamental sequence of commands is a part of almost all arithmetic operations.

Three methods have been used to determine the time to terminate the command pulses. In the first, a logical check is made to verify the correctness of each transfer in all digit positions. If, and only if, this verification is obtained, the gate signal is terminated and the next step is initiated. The occurrence of random undetected errors in arithmetic-unit transfers is extremely unlikely. A significant number of components must be used to construct such a command-signal control unit. For example, if the word length is 40 binary digits, then a 40-digit AND gate must be associated with each pair of registers between which information is to flow.

A second method requires less equipment. A check is made to verify that the level of the gating signal has exceeded by a safety margin that required to operate all gates. If this has occurred, and a given length of time has passed, the gate signal is terminated. A further check that the gate level has been reduced past the "off" level is made prior to the initiation of the next step. A "speed bus" can be used to control the length of time allowed for the check.

In the third method, a reference or "master" toggle (or flip-flop) is used which is identical with all other toggles in the register. The gating signal that transfers information is used to change the state of the "master" toggle that controls this gating time. The completion of its transition is used to allow the termination of the gate signal. The master toggle derives its power from the same source as all the other toggles in the register.

For all three of the systems described, changes in the speed of information transfer, caused, for example, by variations in gate-bus-driver characteristics, need not introduce errors. The over-all speed of operation of the machine will change—it always runs as fast as it can.

16.33. MICROPROGRAMMING

The idea of the subroutine (see Sec. 17) occurred to programmers early in the exploitation of stored-program machines. Certain sequences of instructions occur repeatedly and cyclically in the solution of problems. A similar observation can be made about the sequencing of command steps in the execution of the various instructions. The concept of microprogramming suggests that the machine be designed and constructed in such a way that the individual basic command steps, such as clear the accumulator or left shift one place, are accessible for alteration.[44]

One suggested mechanization is the use of a plugboard by means of which the sequencing of command steps may be controlled. The instruction list of a machine can therefore be altered, depending on the particular wiring present in the plugboard. Once a given set of instructions has been designed into the machine by the wiring of a microprogram plugboard, the machine may be used as a stored-program computer in the normal fashion.

Various alternates to this scheme are possible. For example, the information that defines the sequencing of the commands to make up an instruction may be stored in the main accessible storage unit of the machine. In the simplest use of such a facility, the instruction list of a machine would be fixed throughout the execution of a given

program of instructions for the solution of a problem. It may be desirable, however, to change the instruction repertory of the machine in accordance with the type of calculation required at different parts of certain large problems. The programming complexity made possible by such flexibility may seem frightening, but the probable improvement in machine performance may warrant such an arrangement.

Machines designed to take advantage of the enormous flexibility and generality that the microprogramming concept provides must be slower than machines designed with a fixed instruction repertory. This is so because extra equipment must be driven or extra storage consultations must be made.

References

1. Arant, Gene W.: A Time-sequential Tabular Analysis of Flip-flop Logical Operation, *IRE Trans. on Electronic Computers*, **EC-6**(2): 72–74 (June, 1957).
2. Aufenkamp, D. D., and F. E. Hohn: Analysis of Sequential Machines, *IRE Trans. on Electronic Computers*, **EC-6**(4): 276–284 (December, 1957).
3. Berlin, R. D.: Synthesis of *N*-valued Switching Circuits, *IRE Trans. on Electronic Computers*, **EC-7**(1): 52–56 (March, 1958).
4. Burks, Arthur W., and Hao Wang: The Logic of Automata, part I, *J. Assoc. Computing Machinery*, **4**(2): 193–218 (April, 1957).
5. Burks, Arthur W., and Hao Wang: The Logic of Automata, part II, *J. Assoc. Computing Machinery*, **4**(3): 279–297 (July, 1957).
6. Burks, Arthur W., Don W. Warren, and Jesse B. Wright: An Analysis of a Logical Machine Using Parenthesis-free Notation, *Mathematical Tables and Other Aids to Computation*, **VIII**(46): 53–57 (April, 1954).
7. Campeau, Joseph O.: The Synthesis and Analysis of Digital Systems by Boolean Matrices, *IRE Trans. on Electronic Computers*, **EC-6**(4): 231–241(December, 1957).
8. Copi, Irving M., Calvin Elgot, and Jesse B. Wright: Realization of Events by Logical Nets, *J. Assoc. Computing Machinery*, **5**(2): 181–196 (April, 1958).
9. Davies, D. W.: Switching Functions of Three Variables, *IRE Trans. on Electronic Computers*, **EC-6**(4): 265–275 (December, 1957).
10. Epstein, George: Synthesis of Electronic Circuits for Symmetric Functions, *IRE Trans. on Electronic Computers*, **EC-7**(1): 57–60 (March, 1958).
11. Ercoli, Paolo, and Roberto Vacca: Binary Arithmetic for Discretely Variable Word Length in a Serial Computer, *Commun. Assoc. Computing Machinery*, **2**(4): 13–16 (April, 1959).
12. Flehinger, B. J.: Reliability Improvement through Redundancy at Various System Levels, *IBM J. Research Development*, **2**(2): 148–158 (April, 1958).
13. Flores, Ivan: Reflected Number Systems, *IRE Trans. on Electronic Computers*, **EC-5**(2): 79–82 (June, 1956).
14. Foss, F. A.: The Use of a Reflected Code in Digital Control Systems, *IRE Trans. on Electronic Computers*, **EC-3**(4): 1–6 (December, 1954).
15. Gilchrist, B., J. Pomerene, and S. Wong: Fast Carry Logic for Digital Computers, *IRE Trans. on Electronic Computers*, **EC-4**(4): 133–136 (December, 1955).
16. Ginsburg, Seymour: On the Reduction of Superfluous States in a Sequential Machine, *J. Assoc. Computing Machinery*, **6**(2): 259–282 (April, 1959).
17. Glantz, Herbert T.: A Note on Microprogramming, *J. Assoc. Computing Machinery*, **3**(2): 77–84 (April, 1956).
18. Golay, Marcel J. E.: The Logic of Bidirectional Binary Counters, *IRE Trans. on Electronic Computers*, **EC-6**(1): 1–4 (March, 1957).
19. Harris, Bernard: An Algorithm for Determining Minimal Representations of a Logic Function, *IRE Trans. on Electronic Computers*, **EC-6**(2): 103–107 (June, 1957).
20. Hirschhorn, Edwin: Simplification of a Class of Boolean Functions, *J. Assoc. Computing Machinery*, **5**(1): 67–75 (January, 1958).
21. Hohn, F. E., S. Seshu, and D. D. Aufenkamp: The Theory of Nets, *IRE Trans. on Electronic Computers*, **EC-6**(3): 154–161 (September, 1957).
22. Jeenel, J.: Programs as a Tool for Research in Systems Organization, *IBM J. Research Development*, **2**(2): 105–122 (April, 1958).
23. Jeffrey, Richard C.: Arithmetical Analysis of Digital Computing Nets, *J. Assoc. Computing Machinery*, **3**(4): 360–375 (October, 1956).
24. Joachim, Gertrud S.: Memory Efficiency, *J. Assoc. Computing Machinery*, **6**(2): 172–175 (April, 1959).
25. Kochen, M.: Extension of Moore-Shannon Model for Relay Circuits, *IBM J. Research Development*, **3**(2): 169–186 (April, 1959).

26. Korolev, L. N.: Coding and Code Compression, *J. Assoc. Computing Machinery*, **5**(4): 328–330 (October, 1958).
27. Leichner, Gene H.: Designing Computer Circuits with a Computer, *J. Assoc. Computing Machinery*, **4**(2): 143–147 (April, 1957).
28. Loberman, H., and A. Weinberger: Formal Procedures for Connecting Terminals with a Minimum Total Wire Length, *J. Assoc. Computing Machinery*, **4**(4): 428–437 (October, 1957).
29. Mackay, R. S., and R. MacIntyre: Ternary Counters, *IRE Trans. on Electronic Computers*, **EC-4**(4): 144–149 (December, 1955).
30. Marcus, Mitchell P.: Minimization of the Partially-developed Transfer Tree, *IRE Trans. on Electronic Computers*, **EC-6**(2): 92–95 (June, 1957).
31. Mauchly, John W.: The Advantages of Built-in Checking, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 8–10, 1953, pp. 99–101.
32. Mercer, Robert J.: Micro-programming, *J. Assoc. Computing Machinery*, **4**(2): 157–171 (April, 1957).
33. Miehle, William: Burroughs Truth Function Evaluator, *J. Assoc. Computing Machinery*, **4**(2): 189–192 (April, 1957).
34. Morris, E. F., and T. E. Wohr: Automatic Implementation of Computer Logic, *Commun. Assoc. Computing Machinery*, **1**(5): 14–20 (May, 1958).
35. Muller, D. E.: Application of Boolean Algebra to Switching Circuit Design and to Error Detection, *IRE Trans. on Electronic Computers*, **EC-3**(3): 6–12 (September, 1954).
36. Nelson, E. C.: An Algebraic Theory for Use in Digital Computer Design, *IRE Trans. on Electronic Computers*, **EC-3**(3): 12–21 (September, 1954).
37. Patterson, George W.: Symbolic Methods in the Design of Delay- and Cycle-free Logical Nets, *IRE Convention Record*, part 4, New York, 1954, pp. 58–64.
38. Peterson, W. W., and M. O. Rabin: On Codes for Checking Logical Operations, *IBM J. Research Development*, **3**(2): 163–168 (April, 1959).
39. Shepherdson, J. C.: The Reduction of Two-way Automata to One-way Automata, *IBM J. Research Development*, **3**(2): 198–200 (April, 1959).
40. Sisson, Roger L.: An Improved Decimal Redundancy Check, *Commun. Assoc. Computing Machinery*, **1**(5): 10–12 (May, 1958).
41. Swift, Charles J.: Machine Features for a More Automatic Monitoring System on Digital Computers, *J. Assoc. Computing Machinery*, **4**(2): 172–173 (April, 1957).
42. Urbano, R. H., and R. K. Mueller: A Topological Method for the Determination of the Minimal Forms of a Boolean Function, *IRE Trans. on Electronic Computers*, **EC-5**(3): 126–132 (September, 1956).
43. Wang, Hao: A Variant to Turing's Theory of Computing Machines, *J. Assoc. Computing Machinery*, **4**(1): 63–92 (January, 1957).
44. Wilkes, M. V., and J. B. Stringer: Micro-programming and the Design of the Control Circuits in an Electronic Digital Computer, *Proc. Cambridge Phil. Soc.*, **49**: 230–238 (April, 1953).
45. Wolpe, Harold: Algorithm for Analyzing Logical Statements to Produce a Truth Function Table, *Commun. Assoc. Computing Machinery*, **1**(3): 4–13 (March, 1958).

Section 17

INTRODUCTION TO CODING AND PROBLEM LOGIC

By

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

MICHAEL WOODGER, *Mathematics Division, National Physical Laboratory, Teddington, Middlesex, England.*

CONTENTS

17.1. The Language of Programming 17-2
17.2. Characters or Symbols 17–2
17.2.1. Relational Operators 17-2
17.2.2. Logical Operators 17-2
17.2.3. Sequential-control Operators 17–3
17.2.4. Separators 17–3
17.2.5. Grouping Symbols 17–3
17.3. Items 17–4
17.4. Arithmetic and Boolean Expressions 17–4
17.5. Statements 17–5
17.5.1. Assignment Statements 17–5
17.5.2. Array Statements 17–5
17.5.3. Type Statements 17–5
17.5.4. DO Statements 17–5
17.5.5. GO TO Statements 17–6
17.5.6. Conditional Statements 17–6
17.6. Subscripted Variables 17–6
17.7. Grouping of Statements 17–7
17.8. Indexing 17–7
17.9. Functions 17–7
17.10. Programming Example 17–7
17.11. Structure of Programs 17–8
17.12. Flow Diagrams 17–8
17.13. Machine Language 17–10
17.14. Subroutines 17–11
17.15. Transfer of Control 17–11
17.16. Index Registers 17–11
17.17. Command List 17–12
17.18. Arithmetic Commands 17–13
17.19. Input and Output Commands 17–14
17.20. Program-logic Commands 17–15
17.21. Solution of a Quadratic Equation 17–15
17.22. An Example Using a Subroutine 17–15
17.23. A Scalar-product Example 17–17
17.24. The Scalar-product Example without Index Registers 17–18
17.25. Scalar Product with a Linear Code 17–19
17.26. A Generator for the Scalar-product Linear Code 17–19
17.27. Comparison of the Coding Methods of the Last Four Sections 17–19
17.28. Interpretive Systems 17–21
17.29. A "Mark Transfer" Approach to Subroutine Organization 17–22
17.30. Command Systems 17–23
17.31. Commands of Special Interest 17–24
17.31.1. Add-multiply Command 17–24
17.31.2. Repeat Command 17–24
17.31.3. Shift-jump Command 17–24
17.31.4. Normalize 17–25
17.31.5. Merge 17–25
17.31.6. Sift 17–25
17.31.7. Summarize 17–25
17.32. Compilers 17–25
17.33. An Introduction to ALGOL 60 17–26
17.33.1. Introduction 17–26
17.33.2. Examples 17–29
17.33.3. Syntax 17–31
17.33.4. Syntax of Expressions 17–31
17.33.5. Syntax of Statements 17–34
17.33.6. Syntax of Declarations 17–35
17.33.7. Classification of Basic Symbols 17–36
17.33.8. Example 17–37

The process of solving a problem may be broken up into the following steps: (1) formulation, (2) analysis, (3) programming, (4) coding, (5) computing, and (6) analysis of results. Analysis includes finding a suitable numerical method for the problem, determining how many significant digits are required at each step in the process, and determining in what form the results are to be displayed. This discussion is concerned with the process after such steps have been taken. In particular, programming and coding are the only steps which will be considered in this chapter.

Programming consists of determining the proper sequence of events required to accomplish the computation. Generally, such a process depends upon the numerical methods chosen and the required accuracy, and it may involve certain optimization processes to minimize the number of arithmetic or logical steps required to arrive at the solution.

Coding is the process of translating the program into the language of a particular computer. Whereas programming is usually independent of the computer upon which the problem is to be done, the process of coding varies from computer to computer.

17.1. THE LANGUAGE OF PROGRAMMING

A language suitable for specifying the computational process described above has been evolving in the computer field. This language is not only useful for describing processes which are to be run on a computer but it is a medium of communication relative to computational processes. For example, it can be used to publish computational methods or algorithms.

An ideal language of this type must (1) be readable by those who are interested in computational processes without the need of further explanations, and (2) it should be sufficiently complete so that any computational process can be described in it without difficulty or ambiguity.

Sections 17.2 to 17.9 describe some of the characteristics of such a language. Actually, a somewhat simplified version of an algorithmic language called ALGOL[29] is described. Sections 17.10 to 17.32 describe in some detail command systems and the process of coding for various systems. An introduction to ALGOL 60 is given in 17.33.

17.2. CHARACTERS OR SYMBOLS

The symbols used in statements defining computational processes are usually limited by characteristics of input and output equipment used on computers. In publication the limit is in terms of available type that can be used by the printer. A more extensive list of symbols is presented here than is found on most input or output equipment. The symbols can be classified into groups such as (1) digits, (2) letters, and (3) operational symbols. Operational symbols are further classified as (*a*) arithmetic operators, (*b*) relational operators, (*c*) logical operators, (*d*) sequential operators, (*e*) separators, and (*f*) grouping symbols.

Of these symbols, strings of digits and operation symbols have conventional meanings either well known in mathematics or to be explained in what follows. Individual letters and strings made up of a letter followed by a string of digits and/or letters are labels or identifiers and will be referred to as variables hereafter.

The various operational symbols are given in Table 17.1.

17.2.1. Relational Operators. Relational operators are used with two operands, and for any two given operands the relationship may or may not exist. For example, $a < b$ is either true or false. It may be convenient to interpret a relational operator and its two operands as a Boolean variable, equal to 1 if true and equal to 0 if false.

17.2.2. Logical Operators. There are three logical operators given in Table 17.1, and the operands are Boolean variables. The meaning of the operators is given by the following statements:

$$0 \vee 0 = 0 \qquad 0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1$$
$$1 \wedge 1 = 1 \qquad 0 \wedge 0 = 1 \wedge 0 = 0 \wedge 1 = 0$$
$$\neg 0 = 1 \qquad \neg 1 = 0$$

Table 17.1. Symbols

Category	Symbol	Contextual form	Meaning
Arithmetic	$+$	$a + b$	a plus b
	$-$	$a - b$	a minus b
	$\cdot$	$a \cdot b$	a times b
	$/$	a / b	a divided by b
Relational	$<$	$a < b$	a less than b
	$\leq$	$a \leq b$	a less than or equal b
	$=$	$a = b$	a equal b
	$\geq$	$a \geq b$	a greater than or equal b
	$>$	$a > b$	a greater than b
	$\neq$	$a \neq b$	a not equal to b
Logical	$\neg$	$\neg a$	Not a
	$\vee$	$a \vee b$	a or b
	$\wedge$	$a \wedge b$	a and b
Separators	,	$a =: b, c =: d,$	
	:	$s: a =: b,$	
	;	$a =: b; c + \cdots$	
	:=	$a := b$	a is replaced by b
	=:	$a =: b$	a replaces b
	$_{10}$	$57_{10} - 3$	57×10^{-3}
	.	$a =: b.$	
Grouping	( )	$(a + b) - c =: z$	
	[ ]	$a[i]$	a sub i
	{ }	$\{a =: d\}$	
	↑ ↓	$a\uparrow b\downarrow$	a to the power b

The operator $\wedge$ is called "and" and $\vee$ is called "or." Sometimes, where no ambiguity occurs, some authors use $\cdot$ (times) and $+$ (plus) for $\wedge$ and $\vee$.

17.2.3. Sequential-control Operators. A computation usually consists of a number of arithmetic statements to be performed in a particular order. The sequential-control operators make it possible to specify the order and the conditions under which the various statements are to be executed.

17.2.4. Separators. The symbols , ; and . play the usual punctuation role and the . is also used as a decimal point in the conventional way. The colon (:) is used to separate the name of a statement from the statement itself and to separate conditions from the related statement (see Art. 17.5.6).

The symbols =: and := (which are considered as two single symbols compounded from existing typewriter symbols) are read as "replaces" and "is replaced by," respectively. In most mathematical literature these two symbols are represented by the symbol "$=$". The two new symbols are introduced here to remove the ambiguity and the equality symbol (=) is used only in the relational sense. The symbol $_{10}$ separates the significant figures of a number from its exponent or scale factor.

17.2.5. Grouping Symbols. Parentheses,) and (, are used in the conventional way in mathematical expressions. The square brackets ([and]) are used to enclose subscripts, as illustrated in Table 17.1. This use represents a compromise aimed at facilitating single-line printing of formulas and still retaining, hopefully, ease of reading and interpretation. The brace symbols ({ and }) are used to enclose statements. Note that in ALGOL[29] instead of braces the words "begin" and "end" were used. The up and down arrows (↑ and ↓) are used to enclose exponents. All these symbols may be nested in the conventional way.

17.3. ITEMS

Computation is the processing of information. Units of information occurring in a computation process will be called *items*. One kind of item is called numbers. Numbers may be classified into integers, floating-point numbers, and Boolean numbers. In this classification, fractions and mixed numbers have been grouped with floating-point numbers (numbers given in two parts, a significant-figure part and a scale-factor part, as illustrated in Table 17.2). Note that integers also qualify as floating-point numbers, but they are of sufficient importance to be considered as a separate class.

In this presentation it becomes important to distinguish between items and their names. When one speaks of the variable "x" or of the "credit balance" these are the names of certain numbers occurring in the computation. On the other hand, when one says "3.14" this is the actual number being considered. Following conventional mathematical practice variables (see Sec. 17.2) such as "x," "ABC" will be used as the names of certain items of information. Parentheses will be used to indicate that the enclosed item has become the name of some other item. For example "x" is the name of some item, whereas "(x)" denotes that item whose name is named "x." In a similar way (3.14) denotes that item whose name is 3.14, and a priori, the item may have no numerical relation to the number 3.14.

Note that there is an inconsistency in this notation. For example, "x" is the name of some number but "3" is a number itself. The reason for this inconsistency is that in computations the items most often referred to are variables or numbers and some "principle of minimum effort" dictates that people use the simplest notations for these concepts.

Since variables are the names of other items it is necessary to introduce further conventions in order to make it possible to specify manipulations of the symbols themselves. To this end the string "x" denotes the symbol x itself. Considering the conventions mentioned in the preceding paragraphs, note that "3.14" and 3.14 are identical.

The purpose of the above discussion may be more easily understood in terms of data-processing machines. Such machines store items of information in memory locations which are usually numbered 0, 1, 2, . . . , 9999 (say). A variable x may be stored in location 507. Therefore, both x and (507) serve as names for the number stored in location 507. As another example, suppose that location 603 contains the integer 507, then the variable x has three names: x, (507), and ((603)).

Table 17.2. Numbers

Number	ALGOL representation	Excess-50 representation
507.3	$5073_{10} - 1$	53.5073
15	15	52.1500
0.0167	0.0167	49.1670

17.4. ARITHMETIC AND BOOLEAN EXPRESSIONS

A number or a variable whose range of values is a subset of the rational numbers is an arithmetic expression.

If E and F are expressions then $+E$, $-E$, $E + F$, $E - F$, $E \cdot F$, E/F, $E \uparrow F$ (E to the power F), and (E) are expressions provided they are defined and are not ambiguous.

Some ambiguity in the meaning of possible arithmetic expressions is eliminated by the following evaluation rules: (1) innermost parentheses are evaluated first, (2) within any level of parentheses, multiplications and divisions are done first in order from left to right, then additions and subtractions in order from left to right. The

exponentiation symbols are treated like parentheses. If logical operators occur inside a given level of parentheses first all "$\neg$" are done in order left to right, then all "$\wedge$," followed finally by all "$\vee$." The scope of the "$\neg$" operator is the first item to its right.

Boolean Vector Expressions. The value of an expression E is represented by a sequence of binary bits. The expression $E = E_1 \wedge E_2$ has a 1 bit in each position of E for which the corresponding positions of E_1 and E_2 both contain a 1 bit. The other positions of E contain 0 bits.

The expression $E = E_1 \vee E_2$ has a 1 bit in each position for which there is a 1 bit in the corresponding positions of either E_1 or E_2.

A Boolean vector variable p is said to be "1" if each bit position of p contains a 1 bit. Likewise, p is said to be "0" if each bit position of p contains a 0 bit. This makes it possible to mix Boolean variables and numerical variables. For example, suppose a, b, and c are numerical variables and p and q are Boolean vector variables, then it is possible to write $p \wedge a + q \wedge b =: c$, where c can have the value 0, a, b or $a + b$, depending upon the values of p and q.

This scheme makes it possible to deal with units of information which are different in length or structure from units normally processed by the computer. For example, p may have any grouping of 0's and 1's in its binary positions.

Contrast this discussion with the conventions of ALGOL (Art. 17.33.4).

17.5. STATEMENTS

Statements are classified as follows:

Declarative statements

- Assignment
- Array
- Type

Imperative statements

- DO
- GO TO
- Return
- For

Conditional statements

17.5.1. Assignment Statements. If E is an expression and V is a variable, an assignment statement has the form $E =: V$ or $V := E$. If N is the name of the statement it may be written $N: E =: V$. When such a statement is executed the rational number resulting from the evaluation of the expression E is assigned as the value of the variable V.

17.5.2. Array Statements. Array declarations indicate the number of elements in the array represented by a variable V. For example, $V(10,15)$ indicates that the variable V denotes the elements of a matrix with 10 columns and 15 rows.

17.5.3. Type Statements. Type declarations specify whether a particular variable takes on integral values, Boolean values, or rational values. If not specified, it is assumed that variables take on rational values.

17.5.4. DO Statements. A computation is specified by a number of statements, and some of these statements must be executed in a specific order. DO statements, GO TO statements (see Art. 17.5.5), make it possible to specify the order of execution of the various statements.

The form of a DO statement may be $N5$ and in context may appear as $\cdots + b =: c$, $N5$, $x - \cdots$. This implies there is a statement whose label is $N5$. The scope of the DO is from the label $N5$ to the first semicolon (;) or period (not inside braces). The termination of the DO process may depend upon dynamic circumstances in that it may itself contain alternative statements and have more than one exit. Normally the

exit is a "return" to the higher-level program. In the example above, continuation is with the statement starting with "x — · · · ".

17.5.5. GO TO Statements. A GO TO statement marks the termination of the current process and specifies which statement is to be executed next. The form of a GO TO statement is $N5$. The next statement to be executed is labeled $N5$.

17.5.6. Conditional Statements. Alternative or conditional statements are statements of any of the above types preceded by a condition which must be met in order for the statement actually to be executed. For example, suppose that S is a statement of one of the above types, then an alternative statement may appear as

$$a > b{:}S;\ \cdots \qquad \text{or} \qquad \text{If } a > b \text{ then S;}$$

in which case the execution of the statement S occurs only if a is greater than b at the time of the computation. In order to arrange to execute one of several statements depending upon certain conditions, the following convention is used. If a statement is executed, the next statement to be executed is that following the next semicolon (;) or period. For example, consider the following compound statement

$$c > d{:}S,R,T;U,\ \ldots .$$

If at the time of the computation c is greater than d, statements S, U, etc., will be executed; whereas if c is less than or equal to d, statements R, T, U, etc., will be executed.

For the ALGOL form of conditional statements see Art. 17.33.5.

Table 17.3. Examples of Elementary Statements

Type of statement	Example	Remarks
Assignment	$a + b =: c$,	The sum of the numbers called a and b replaces the number called c
Array	$a(10,15)$,	There are 150 elements called a in the form of a matrix with 10 columns and 15 rows. The individual elements will be referred to by the subscripted variable $a[i,j]$, say
Type	j(I,5)	There are five items named j and they assume integral values only. In many computers this indicates that fixed-point arithmetic may be used instead of floating-point. In any case if this variable is about to be assigned a nonintegral value the number must be rounded to the nearest integer before assignment
DO	N, or N;	DO the statement named N. The scope of the DO is from the label N to the next following semicolon or period (in the dynamic sense and at proper brace level)
GO TO	N.	Execute statement N next
Conditional	$c > d{:}N,M;P,\ \ldots$	If c greater than d execute N and continue with P. Otherwise, execute M and continue with P. The symbol $>$ may be replaced by any relational symbol

17.6. SUBSCRIPTED VARIABLES

If V is a variable and E is a generic notation for expressions then $V[E,E,\ \ldots\ ,E]$ is a subscripted variable. If any of the expressions do not have integral values they have to be rounded to the nearest integer.

17.7. GROUPING OF STATEMENTS

If S is a generic notation for statements then $\{S,S,\ \ldots\ ,S\}$ is also a statement. This device, with the possibility of nesting such symbols, makes it possible to extend the scope of DO statements or to do complex operations as a part of conditional statements.

17.8. INDEXING

In computational mathematics it is well known that the statement $i = 1(1)10$ means that i is to take on the values 1, 2, 3, . . . , 10. Likewise, $x = 0.1(0.01)5$ means that x takes on the values 0.1, 0.11, 0.12, . . . , 4.99, 5. In some branches of mathematics it is the practice to write sequences in the form

$$\left\{\ \ldots\right\}_{i=1}^{100}$$

or in the evaluation process of a definite integral to write

$$\int_a^b f(x)\,dx = F(x)\Big|_{x=a}^{x=b}$$

Therefore, if a statement is to be evaluated for a sequence of parameters (such as an index) it can be written in the form

$$\{S\}i = 1(1)10,\ \ldots$$

which means that statement S is executed 10 times with i taking on the values 1, 2, 3, . . . , 10.

17.9. FUNCTIONS

A function is a rule or law which for each "point" in the range space associates a point or points in the value space. The point in the range space is called the argument of the function. If for each argument there is a single point in the range space the function is said to be single-valued.

Functions which occur in computational programs tend to be of two types. One type is rational-valued functions of one or more rational arguments. The other type might be called "mappings" of certain information from auxiliary devices into the computer or from the computer to the auxiliary devices. For example, the printing of a report (results of a computation) is a mapping of certain information in the computer onto printed pages.

Functions such as $\sin(x)$ or $\text{out5}(a,b,c)$ can be defined by use of the appropriate statements and then used in the specification of the computation.

17.10. PROGRAMMING EXAMPLE

The Evaluation of a Polynomial. A polynomial $p(x)$ may be written in the form $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$. However, the most efficient method of evaluation involves expressing the polynomial in the form

$$p(x) = x(x(x(\cdots(xa_n + a_{n-1}) + a_{n-2})\cdots) + a_0$$

This can be expressed in terms of an iterative formula as follows:

$$p_i(x) = x(p_{i-1}(x)) + a_{n-i}$$

where $p_0(x) = a_n$,

$$i = 1(1)n, \text{ and } p_n(x) = p(x)$$

This last form is a good computational algorithm for the evaluation of a polynomial

with given coefficients. Using the techniques of the preceding sections a more detailed computational algorithm is as follows:

$$s0\colon\ 0 =: p,\ \{x \cdot p + a[n - i] =: p\}\ i = 0(1)n,$$
$$s1\colon \text{ subsequent computation}$$

In the above example the quantity p is initially made equal to zero. After the execution of the second part of the statement $(i = 0)$ p is equal to $a[n]$. A second execution produces $x \cdot a[n] + a[n - 1]$. When the second part has been executed for all the values of i, p has assumed the value of the polynomial.

17.11. STRUCTURE OF PROGRAMS

A program may consist of a single sequence of statements which are to be executed only once. Such a program is said to be a *linear program*. Alternative statements may be used to provide branches in the logic of the program. If, for each set of conditions to be tested by the conditional parts of the alternative statements, the resulting program is linear, then the program is said to have the structure of a *tree*. If the structure of the program is such that some statements are executed more than one time, the program is said to contain cycles. Generally, a practical program has a combined tree and cyclic structure. Cycles may be *simple* if a sequence of statements is repeated a fixed number of times. On the other hand, there may be cycles within cycles combined with tree structures in a complicated way.

To help the programmer visualize and understand the computation process a technique of describing the structure of the program in terms of a so-called *flow diagram* has been developed.

17.12. FLOW DIAGRAMS

The program for a computation is usually expressed in a *flow diagram*. This is a diagram or map illustrating the sequence of steps necessary to accomplish the computation. The elementary entries on such a flow diagram may be actual statements or even machine commands, or they may represent sequences of statements called *routines* or *subroutines*. In the following material three methods of presenting programs in flow-diagram form will be given. These three methods differ only in the symbols used. In all cases it is a matter of indicating the logical interconnection of the various steps which must be done to accomplish the computation.

Most calculators do only one operation at a time so that the problem is accomplished by executing a single time sequence of commands. However, many problems involve iterative formulas or simple formulas that are repeatedly applied to a sequence of numbers. Thus from the spatial point of view the commands which accomplish the computation may have a very complicated pattern of interconnection.

In order to specify the computation process the flow diagram must specify both this spatial interconnection and time dependence. A very complicated problem is usually presented in terms of symbols at a number of different levels. There may be a master flow diagram in which most of the units are collections of subroutines accomplishing quite erudite mathematical operations (line integrals, for example), the subroutines represent the next lower level, and finally, the actual machine commands represent the most detailed language used in the flow diagram.

Arithmetic operations are specified as units in a flow diagram using symbols very similar to that used in the actual mathematical formulas. An arrow can be read as "replaces"; $a \rightarrow$ b means a replaces b. Note that in ALGOL the arrow is replaced by the symbol : =. Branch points are used in flow diagrams where there is a choice of two sequences of operations to perform. Whenever there are more than two choices to pursue the choice is determined by a *distributor* or *switch*. Whenever two sequences of commands come together and continue as a single sequence the first operation common to both sequences is called a *junction*. Sometimes status statements appear indicating the condition of certain quantities. These frequently occur

after branch points to identify and distinguish the two branches. Conditional branches may be used where the choice for continuing depends upon some variable quantity in the problem.

Figure 17.1 shows a box flow diagram for the problem of finding the roots of the quadratic equation $ax^2 + bx + c = 0$. At the end of the process an exit is indicated, permitting the computation to continue with the next part of the problem. If this was the last operation to be performed the E would be replaced by a "halt."

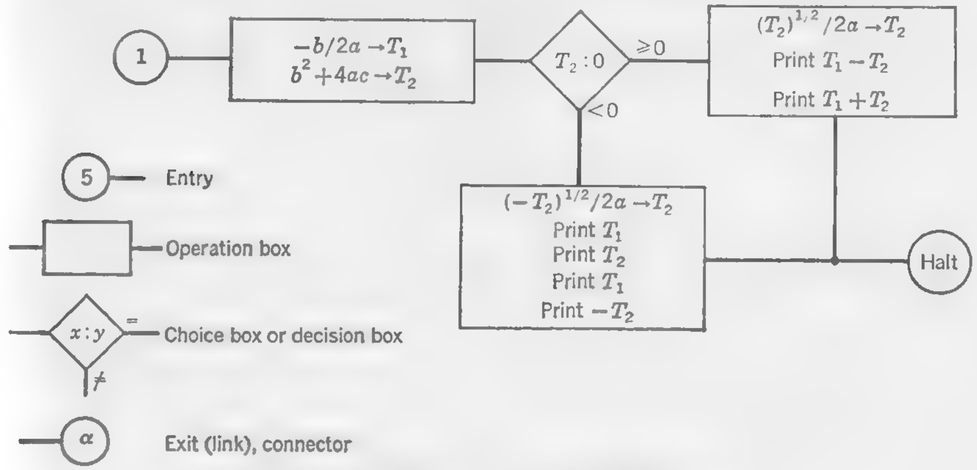


FIG. 17.1. Flow diagram for quadratic equation.

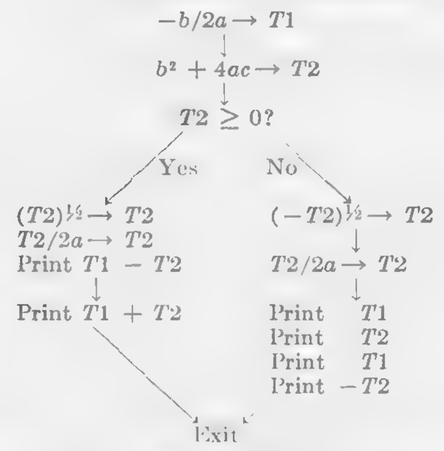


FIG. 17.2. Columnar flow diagram for quadratic equation.

Figure 17.2 shows the flow diagram for the solution of the same quadratic equation in a form not requiring so much drafting effort.

Figure 17.3 shows a typed form of the same flow diagram. It is interesting to consider the possibility that a computer with alphanumeric output might type its own flow diagrams.

Various conventions in the use of flow diagrams have been established at different computing laboratories. Schemes similar to that of Fig. 17.1 are in widest use. In some systems arrows are only from left to right or from top to bottom, and in others arrows may go in any direction. The various boxes have distinct names, as indicated in Figs. 17.1 and 17.4. The bottom of Fig. 17.4 shows the use of fixed and variable connectors.

```
(1) −b/2a → T1
    b² − 4ac → T2
    T2: 0
      T2 ≥ 0:
        (T2)½ → T2
        T2/2a → T2
        Print T1 − T2
        Print T1 + T2
        To (2)
      T2 < 0:
        (−T2) ½ → T2
        T2/2a → T2
        Print   T1
        Print   T2
        Print   T1
        Print  −T2
(2) Exit
```

FIG. 17.3. Tabular flow diagram for quadratic equation.

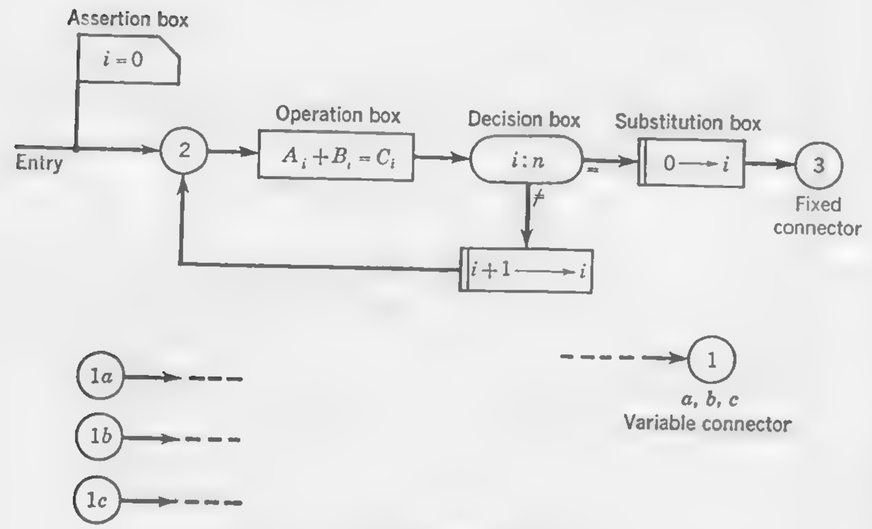


FIG. 17.4. Flow-diagram example.

17.13. MACHINE LANGUAGE

Each computing machine has a limited list of operations that it can perform. These operations are characterized by the command list for that particular machine. There are many computers which are able to do only fixed-point arithmetic in the binary number system. Others do fixed-point decimal arithmetic as their elementary operations. A few are capable of floating-point arithmetic in either binary or decimal number systems. Besides arithmetic operations all computers have commands which make it possible to organize sequences of commands, repeating them an appropriate number of times or using them as an entity in a larger sequence of commands. Each computer also has input and output commands which provide for communication with the outside world. Such input and output commands may not provide all the desired facilities with respect to output format or the form in which input data are supplied to the computer. Sequences of commands may be established and then used as units to provide for certain patterns of input and output.

17.14. SUBROUTINES

The groups of commands used as units as described in the preceding paragraph are called *subroutines*. If the built-in arithmetic operations are fixed-point binary then the group that uses the machine will program subroutines which accomplish floating-point binary operations and binary-to-decimal and decimal-to-binary conversion routines for each kind of number. Sooner or later multiple precision routines will be developed, and perhaps even routines for doing arithmetic using complex numbers. Subroutines have been developed for other than arithmetic operations, such as sorting, collating, and printing.

17.15. TRANSFER OF CONTROL

Problems to be done on automatic computers are usually characterized by the fact that they involve application of the same process to a sequence of numbers. For example, a problem in numerical integration may involve the application of an iterative formula to a sequence of numbers to produce an extrapolated value. The process "marches" in the sense that the extrapolated value is now used to produce a new extrapolated value, and so forth. Or in another case a formula may be evaluated for many different values or combinations of values of the parameters.

The evaluation of the formula may itself involve iterative processes (Newton's method for solving equations, for example); so that instructions for doing the computation must involve not only the sequence of arithmetic operations that are to be performed, but also instructions about how many times certain subsequences of instructions are to be performed. Furthermore, the number of iterations of these subsequences may depend upon rates of convergence that cannot be easily known in advance.

Once the particular computation is completed it is usually necessary to move on to the next set of arguments and finally to terminate the process after a certain range of arguments has been covered.

Transfer of control commands provide the links to connect together the various sequences of arithmetic operations. Unconditional or absolute transfer of control can cause the computer to repeat a certain sequence indefinitely. Conditional transfer of control commands make it possible to repeat sequences of commands either a prescribed number of times, or numbers of times which depend upon some of the variables involved in the computation. Conditional transfer of control may depend upon the signs of numbers, whether they are zero or not, whether overflow has occurred, whether certain switches have been set, or whether input or output processes have terminated.

In most calculators the conditional transfers of control are made to depend upon the contents of an arithmetic register called the accumulator. In a few cases they can depend upon numbers stored any place in the operating memory.

17.16. INDEX REGISTERS

Whenever a sequence of commands is to be applied to a sequence of numbers the addresses of the commands which refer to operands or perhaps to results must be modified as the computation progresses. There is one exception in that the operands may be moved so that they are always processed out of the same addresses.

In some machines such modification of addresses must be done in the accumulator where all other arithmetic operations are done. In some computers the addresses in the command may be modified as they are executed by the addition of the contents of another register called a *B* or index register. The contents of the index register may be changed from time to time, thus allowing the program to be applied to a sequence of numbers without modification of the actual commands in the code.

In the simplest form there is one index register. Each command may or may not refer to the index register. If it does the address is added to the contents of the index

register before the command is executed. The command list contains commands which can set the index register to a given value, decrease it by unity, and provide for transfer of control until the index register is decreased to zero.

In more elaborate arrangements there may be several index registers and it may be possible to change them other than by unity; they may be increased as well as decreased. Limits may be set for each one so that processes may be terminated when a particular index register is incremented to or beyond its limit.

17.17. COMMAND LIST

In order to explain coding and programming, a list of commands such as might be found in some particular computer will be presented here. The command list presented here is not the command list of any actual computer, but at least one computer has been programmed to operate in an interpretive mode as if this were essentially its command list.

The commands will be discussed in three groups, (1) arithmetic commands, (2) input and output commands, and (3) program logic commands. Each command consists of three parts: (*a*) an operation code, (*b*) an address, and (*c*) an index register designation. For the purposes of this chapter it will be assumed that the memory stores 1,000 numbers or words (commands) addressed by the numbers 000 to 999. It will be assumed that there are nine index registers numbered 1, 2, 3, . . . , 9. The digit 0 in the index-register position indicates that no index register is involved in that command.

An index register stores a three-digit decimal number. Whenever the index digit is not zero the contents of the corresponding index register is added (modulo 1,000) to the address portion of the command before execution. This resulting address is referred to as the effective address.

The following three sections will present the three groups of commands. This will be followed by an explanation of index registers and by coding examples to illustrate the various points of interest.

Table 17.4. Arithmetic Commands

Command	Mnemonic code	Effect
1. CLEAR ADD	*CA*	Transfers the contents of the effective address to the accumulator
2. CLEAR SUBTRACT	*CS*	Transfers the negative of the contents of the effective address to the accumulator
3. ADD	*AD*	Adds the contents of the effective address to the contents of the accumulator and leaves the result in the accumulator
4. SUBTRACT	*SU*	Subtracts the number from the accumulator
5. MULTIPLY	*MU*	Multiplies the contents of the effective address by the contents of the accumulator and leaves the result in the accumulator
6. DIVIDE	*DL*	Divides the contents of the accumulator by the contents of the effective address and leaves the result in the accumulator
7. ABSOLUTE VALUE	*AV*	The absolute value of the contents of the accumulator replaces the contents of the accumulator
8. SQUARE ROOT	*SR*	If the contents of the accumulator was nonnegative its square root is left in the accumulator
9. NEGATE	*NE*	Change the sign of *A*
10. STORE	*ST*	The contents of the accumulator replaces the contents of the effective address (it is also left in the accumulator)

17.18. ARITHMETIC COMMANDS

Numerical information is stored in the main memory (1,000 addresses) or in the accumulator (1 number). The results of all the arithmetic commands are left in the accumulator, and in the case of the store command the result is also placed in a specified address in the memory. Word length is not pertinent to this discussion; it can be assumed to be 10 decimal digits and a two-digit exponent, making it possible to handle numbers whose absolute value lies between 10^{-50} and 10^{50}. Arithmetic operations may be in decimals, may be programmed in a fixed-point arithmetic unit, or may even be done in the binary number system.

If any arithmetic operations produce answers greater than 10^{50} or less than -10^{50} the computer halts and gives an alarm. If the operand for the square root is negative the computer halts and gives an alarm. If any answer lies between 10^{-50} and -10^{-50}, then 10^{-50} is left in the accumulator. Thus 10^{-50} plays the role of zero in the number

Table 17.5. Input and Output Commands

Command type	Mnemonic code	Effect
1. Input	*TI*	Computer halts and waits for the operator to type a number which will be transferred to the accumulator
2. Card input	*CI*	Numbers are read from punched cards and placed in the effective address and those that follow it until a "flag" appears on the cards. The accumulator is not disturbed
3. Magnetic-tape input	*MI*	The next 100-word block is read from the tape unit and stored in the effective address and the 99 addresses which follow it
4. Type and tab	*TT*	Types the contents of the accumulator and operates the tab key on the typewriter. The contents of the accumulator is undisturbed
5. Type and carriage return	*TR*	As in *TT* except the carriage-return key operated in place of tab
6. Type address	*TA*	Types the effective address. Thus, if the index digit $\neq 0$ the contents of the corresponding index register is added to the address before typing.
7. Space	*SP*	If the digits of the specified address are a_1, a_2, and a_3 in order of significance then a_2 carriage returns (and line feeds) are performed followed by a_3 tabs
8. Card output	*CO*	If the index digit of the command is zero the word specified by the address portion of the command is punched onto a card and the card is flagged with a special punch. If the index digit is not zero then the contents of the designated index register is added to the effective address and the contents of this memory address is punched onto the card. The index register is increased by its increment and another number is punched. This process continues until the index register reaches its limit. The last card punched is "flagged"
9. Magnetic-tape output	*MO*	A block of 100 words starting at the effective address is recorded in a forward direction on the tape. This erases information previously recorded in that block
10. Magnetic-tape reverse search	*MR*	The magnetic tape is reversed the number of blocks specified by the effective address. No recording or reading of data is done
11. Magnetic-tape forward search	*MF*	As in reverse search, but in forward direction

system. If the number $A \times 10^b$ is subtracted from itself the answer is written as 10^{-50}. All answers are normalized; that is, the significant figures are shifted left until there is a digit in the most significant position, and the exponent is adjusted subject to the conditions stated above.

17.19. INPUT AND OUTPUT COMMANDS

The input and output commands account for manual input of numbers, printed-page output, recording on magnetic tape or punched cards, and input from punched cards or magnetic tape. For the purposes of this discussion the magnetic tape can be considered as auxiliary memory and the punched cards as a means of reading in data prepared at other locations.

Transfers to and from magnetic tape are in terms of 100-word blocks. Thus the commands need to specify only the location of the initial word of the block. Likewise, input from cards is in terms of variable-length blocks. The end of the block is marked by a "flag" and only the address of the initial item need be specified in the command.

Table 17.6. Program-logic Commands

Command	Mnemonic code	Effect
1. Transfer control	*TC*	The source of commands becomes the effective address and those that follow it
2. Transfer on negative	*TN*	If the contents of the accumulator is negative the effect is the same as for *TC*. Otherwise the source of command continues in sequence
3. Transfer on positive	*TP*	Similar to *TN* except positive instead of negative. Note that "zero" = 10^{-50} is positive
4. Mark-place transfer	*MT*	The contents of the mark-place registers is shifted $M_j \rightarrow M_{j+1}$ for $j = 3, 2$, and 1; and the normal location of next command is placed in M_1; and the effective address becomes the location of next command as in *TC*
5. Return transfer	*RT*	The contents of M_1 becomes the location of next command and the contents of the mark-place registers is shifted $M_{j+1} \rightarrow M_j$ for $j = 1, 2, 3$
6. Increment index register	*II*	Let B be the contents of the index register specified by the index digit of the command, D its increment, and L its limit. $B + D \rightarrow B$ (mod 1,000). If $B + D$ is not more than L the next command is specified by the address of the command. Otherwise commands are executed in sequence
7. Decrement index register	*DE*	If B, D, and L are as in (6) then $B - D \rightarrow B$ (mod 1,000). If $B - D$ is not less than L the next command is specified by the address in the command. Otherwise commands are executed in sequence
8. Set index register	*SI*	If a is the address in the command and B is the contents of the index register specified in the command, the effect of this command is $a \rightarrow B$
9. Set index increment	*SD*	$a \rightarrow D$ where a and D are as described above
10. Set index limit	*SL*	$a \rightarrow L$ where a and L are as described above
11. Halt	*HA*	Computer stops. If the go switch is operated the computer continues as if this were a transfer-of-control command

17.20. PROGRAM-LOGIC COMMANDS

The program-logic commands provide for the termination of iterative processes, for the interconnection of groups of subroutines, and for other organizational details of the complete program.

17.21. SOLUTION OF A QUADRATIC EQUATION

In Fig. 17.1 the solution of a quadratic equation was shown in flow-diagram form. This example will now be coded in the computer language presented in the preceding paragraphs. First it is necessary to assign storage. This is indicated in the following table:

Location	*Contents*
500	*a*
501	*b*
502	*c*
503	"2"
504	"4"
505	*T*1
506	*T*2

In the initial pass at coding the problem it usually pays to use symbolic addresses and assign storage only at the very end. To illustrate this the symbols in the above table under the heading "contents" will be used as symbolic addresses. This has a further advantage in that its mnemonic character will make it easier for the reader to follow.

If the roots are real they will be printed on the same line. If the roots are complex then four numbers will be printed, the real part followed by plus the imaginary part, and then the real part again followed by minus the imaginary part. Note that this program will not work if the coefficient a is too near to zero.

The index digit is always zero in the commands of this code since there is no occasion to make use of index registers.

The example of Table 17.7 has the structure of a "tree" in that there are branch points but no cycles. Conditional transfer-of-control commands are illustrated (command location 012) as well as unconditional transfer of control (023). The next example illustrates the use of subroutines, index registers, and cycles.

17.22. AN EXAMPLE USING A SUBROUTINE

The problem is that of solving an algebraic equation for a real root using Newton's method. It should be emphasized that blind use of Newton's method in the manner coded here may not produce answers because the derivative may turn out to be nearly zero at some point, or the method may produce a sequence of points which repeat instead of converging to a root.

Let the degree of the equation be n and suppose that the coefficients are stored in addresses $a + i$ where $i = 0, 1, \ldots, n$ and the coefficient in a is the constant term. Newton's formula is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

The evaluation of $f(x_n)$ and $f'(x_n)$ will be done by subroutines. $f(x)$ can be written

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

but in order to minimize the number of arithmetic operations it will be evaluated in the form

$$a_n \cdot * f_n \qquad f_j = x f_{j+1} + a_j \qquad j = n - 1, n - 2, \ldots, 1, 0$$

Table 17.7. Solution of a Quadratic Equation

Flow diagram	Location	Command			Remarks
(1) $-b/2a \rightarrow T1$	000	*CS*	b	0	$-b \rightarrow A$
	001	*DI*	a	0	$A/a \rightarrow A$
	002	*DI*	2	0	$A/2 \rightarrow A = -b/2a$
	003	*ST*	*T*1	0	$A \rightarrow T1$
$b^2 - 4ac \rightarrow T2$	004	*CS*	4	0	"-4" $\rightarrow A$
	005	*MU*	a	0	$Aa \rightarrow A$
	006	*MU*	c	0	$Ac \rightarrow A$
	007	*ST*	*T*2	0	$A \rightarrow T2$
	008	*CA*	b	0	$b \rightarrow A$
	009	*MU*	b	0	$b^2 \rightarrow A$
	010	*AD*	*T*2	0	$A + T2 \rightarrow A = b^2 - 4ac$
	011	*ST*	*T*2	0	$A \rightarrow T2$
$T2$?					
$T_2 \geq 0$:	012	*TN*	024	0	$A < 0$ to 024
$(T2)^{1/2}/2a \rightarrow T2$	013	*SR*	000	0	$\sqrt{A} \rightarrow A = \sqrt{b^2 - 4ac}$
	014	*DI*	2	0	$A/2 \rightarrow A$
	015	*DI*	a	0	$A/a \rightarrow A$
	016	*ST*	*T*2	0	$A \rightarrow T2$
Print $T1 - T2$	017	*CA*	*T*1	0	$T1 \rightarrow A$
	018	*SU*	*T*2	0	$-T2 + A \rightarrow A$
	019	*TT*	000	0	Prints $T1 - T2$
Print $T1 + T2$	020	*CA*	*T*1	0	$T1 \rightarrow A$
	021	*AD*	*T*2	0	$A + T2 \rightarrow A$
	022	*TR*	000	0	Prints $T1 + T2$ and carriage returns
To (2)	023	*TC*	037	0	To halt
$T2 < 0$:					
$(-T2)^{1/2} \rightarrow T2$	024	*CS*	*T*2	0	$-T2 \rightarrow A$
	025	*SR*	000	0	$\sqrt{A} \rightarrow A = i\sqrt{b^2 - 4ac}$
$T2/2a \rightarrow T2$	026	*DI*	2	0	$A/2 \rightarrow A$
	027	*DI*	a	0	$A/a \rightarrow A$
	028	*ST*	*T*2	0	$A \rightarrow T2$
Print $T1$	029	*CA*	*T*1	0	$T1 \rightarrow A$
	030	*TT*	000	0	Prints $T1$
Print $T2$	031	*CA*	*T*2	0	$T2 \rightarrow A$
	032	*TT*	000	0	Prints $T2$
Print $T1$	033	*CA*	*T*1	0	$T1 \rightarrow A$
	034	*TT*	000	0	Prints $T1$
Print $-T_2$	035	*CS*	*T*2	0	$-T2 \rightarrow A$
	036	*TR*	000	0	Prints $-T2$ and C.R.
(2) Halt	037	*HA*	000	0	Halts

Table 17.8*a*. Function Evaluation

Flow diagram	Location	Command			Remarks
(1) $n \rightarrow j$	000	*SB*	$a + n$	1	"$a + n$" becomes the contents of the first index register
	001	*SL*	a	1	"a" becomes the limit of index register number 1
	002	*SD*	1	1	the increment is set to "1"
$a_n \rightarrow f_n$	003	*CA*	000	1	$a_n \rightarrow A$
	004	*DE*	005	1	$j - 1 \rightarrow j$
(2) $xf_{j+1} + a_j \rightarrow f_j$	005	*MU*	x	0	$xA \rightarrow A$
	006	*AD*	000	1	$A + a_{j-1} \rightarrow A$
$j - 1 \rightarrow j$ to (2)	007	*DE*	005	1	$j - 1 \rightarrow j$, $j \geq a$ go to 005
(3) Exit	008	*RT*	000	0	$f(x) = A$, return transfer

Table 17.8*b*. Derivative Evaluation

Flow diagram	Location	Command			Remarks
(1) $n \rightarrow j$	009	*SB*	$a + n$	1	See 000 above
	010	*SD*	1	1	
	011	*CA*	n	0	$n \rightarrow A$
	012	*ST*	*T*1	0	$A \rightarrow T1 = n$
$na_n \rightarrow f'_n$	013	*MU*	000	1	$Aa_n \rightarrow A$
	014	*ST*	*T*2	0	$A \rightarrow T2 = na_n$
	015	*SL*	a1	1	Sets limit for first index register
	016	*DE*	017	1	$j - 1 \rightarrow j$
(2) $xf'_{j+1} + ja_j \rightarrow f'_j$	017	*CA*	*T*1	0	$T1 \rightarrow A$
	018	*SU*	"1"	0	$A -$ "1" $\rightarrow A$
	019	*ST*	*T*1	0	$A \rightarrow T1 = j - 1$
	020	*MU*	000	1	$ja_j \rightarrow A$
	021	*ST*	*T*3	0	$A \rightarrow T3$
	022	*CA*	*T*2	0	$T2 \rightarrow A$
	023	*MU*	x	0	$xA \rightarrow A = xf'_{j+1}$
	024	*AD*	*T*3	0	$A + T3 \rightarrow A$
	025	*ST*	*T*2	0	$A \rightarrow T2 = f'_j$
(3) $j - 1 \rightarrow j \geq a + 1$ to (2)	026	*DE*	017	1	$j - 1 \rightarrow j \geq a + 1$ to 017
(4) Exit	027	*RT*	000	0	Return transfer

Similarly, the derivative will be evaluated in the form

$$na_n \rightarrow f'_n \qquad f'_j = xf'_{j+1} + ja_j \qquad j = n - 1, n - 2, \ldots, 2, 1$$

Subroutines to evaluate the function and the derivative are shown in Tables 17.8*a*. and 17.8*b*. The main routine is shown in Table 17.9.

Table 17.9. Newton's Method Applied to a Polynomial

Flow diagram	Location	Command			Remarks
(1) $0 \rightarrow x$	028	*CA*	"0"	0	"0" $\rightarrow A$
	029	*ST*	x	0	$A \rightarrow x$
(2) $f'(x)$	030	*MT*	009	0	$f'(x) \rightarrow T2$
$f(x)$	031	*MT*	000	0	$f(x) \rightarrow A$
	032	*NE*	000	0	$-f(x) \rightarrow A$
$-f(x)/f'(x) + x_n \rightarrow x_{n+1}$	033	*DI*	*T*2	0	$A/T2 \rightarrow A$
	034	*ST*	*T*1	0	$x_{n+1} - x_n \rightarrow T1$
	035	*AD*	x	0	$A + x_n \rightarrow x_{n+1}$
	036	*ST*	x	0	$A \rightarrow x_{n+1}$
$\lvert x_{n+1} - x_n \rvert - b > 0$:	037	*CA*	*T*1	0	
	038	*AV*	000	0	$\lvert x_{n+1} - x_n \rvert$
	039	*SU*	b	0	$A - b \rightarrow A$
Go to (2);	040	*TP*	030	0	$A > 0$ go to 030
	041	*CA*	x	0	$x \rightarrow A$
Print x	042	*TR*	000	0	Type x and carriage return
Halt	043	*HA*	000	0	Halt

17.23. A SCALAR-PRODUCT EXAMPLE

Assume that the numbers x_i are stored in addresses $100 + i$, and y_i in addresses $300 + i$, for $i = 1(1)100$. Suppose that $S = x_1y_1 + x_2y_2 + \cdots + x_{100}y_{100}$ in address 299 is required. Table 17.10 shows a cyclic-code making use of an index register to accomplish the computation.

The basic operation is given by the following iterative formula:

$$0 \rightarrow S_0$$
$$S_i = S_{i-1} + x_i y_i \qquad i = 1(1)100$$

Inspection of Table 17.10 shows that 506 commands are executed to accomplish the computation. This same example is used to show various approaches to coding

Table 17.10. Cyclic Code for a Scalar Product

Flow diagram	Location	Code	Remarks
(1) $0 \rightarrow S$	000	*CA* 011 0	
	001	*ST* 299 0	
$i_0 = 1$	002	*SB* 001 1	Establishes index register
$\Delta i = 1$	003	*SD* 001 1	
$i_L = 100$	004	*SL* 100 1	
(2) $S + x_i y_i \rightarrow S$	005	*CA* 100 1	Adds another product
	006	*MU* 300 1	
	007	*AD* 299 0	
	008	*ST* 299 0	
$i + 1 \rightarrow i$, if $i \leq i_L$ to (2)	009	*II* 005 1	Increment i, finished?
Halt	010	*HA* 000 0	Yes, halt
	011	zero	

problems, and finally in Sec. 17.27 the characteristics of the various approaches are summarized.

17.24. THE SCALAR-PRODUCT EXAMPLE WITHOUT INDEX REGISTERS

If there are no index registers and a cyclic code is used, then the addresses of some of the commands must be modified.

Table 17.11. Scalar Product without Index Registers

Flow diagram	Location	Command	Remarks
$0 \rightarrow S$	000	*CA* 017	
	001	*ST* 299	
$S + x_i y_i \rightarrow S$	002	*CA* 101	Adds another product
	003	*MU* 301	
	004	*AD* 299	
	005	*ST* 299	
$i + 1 \rightarrow i$	006	*CA* 002	Modifies addresses
	007	*AD* 015	
	008	*ST* 002	
	009	*CA* 003	
	010	*AD* 015	
	011	*ST* 003	
$i > i_L$?	012	*SU* 016	Test limit
No: to 002	013	*TN* 002	
Yes: Halt	014	*HA* 000	
	015	00 001	Address increment
	016	*MU* 401	Limit
	017	zero	

17.25. SCALAR PRODUCT WITH A LINEAR CODE

Assumed that all the commands take the same length of time. The index-register code (Table 17.10) requires the execution of 506 commands to accomplish the computation, whereas the code of Table 17.11 requires the execution of 1,203 commands. Thus the index-register system is twice as fast on the basis of the above assumption.

A linear code can accomplish the above computation with the execution of 400 commands. The only way to improve substantially on this speed is to have an

Table 17.12. Scalar Product with a Linear Code

Flow diagram	Location	Commands	Remarks
$x_1 y_1 \rightarrow S$	000	*CA* 501	
	001	*MU* 701	
	002	*ST* 499	
$x_2 y_2 + S \rightarrow S$	003	*CA* 502	
	004	*MU* 702	
	005	*AD* 499	
	006	*ST* 499	
$x_3 y_3 + S \rightarrow S$	007	*CA* 503	
. . .	. . .	. . .	
$x_{100} y_{100} + S \rightarrow S$	395	*CA* 600	
	396	*MU* 800	
	397	*AD* 499	
	398	*ST* 499	
Halt	399	*HA* 000	

arithmetic unit in which it is possible to accumulate products without intermediate storage of partial results in the memory.

Since 400 commands are required to accomplish the computation, the data must be relocated. The x_i are in locations $500 + i$, and the y_i are in $700 + i$. The result is to appear in 499.

17.26. A GENERATOR FOR THE SCALAR-PRODUCT LINEAR CODE

As indicated above, the linear code is the fastest code but uses up the most memory space and requires more time for input. If memory space is not a problem the input information can be reduced to reasonable size by devising a generator code which is usually cyclic in character and can produce the linear code. If the calculator which is to be used has a magnetic-tape unit the generated code may be semipermanently stored on magnetic tape.

The code in Table 17.13 makes use of index registers in an interesting way to generate a linear code which does not use index registers.

Note that the number of components in the vectors may be changed by changing the command in location 804.

17.27. COMPARISON OF THE CODING METHODS OF THE LAST FOUR SECTIONS

In Table 17.14 a comparison of the various methods is given. If it is assumed that all commands take the same length of time for execution the linear code is the fastest way to accomplish the computation, but it uses the most memory space and poses an input problem. The input problem is solved by use of the generator. Somewhat more difficult to compare is the difference in competence of the coding personnel that is required by the various methods.

For this particular example, the cyclic index-register code (Sec. 17.23) shows up

Table 17.13. Generator for Scalar-product Routine

Flow diagram	Location	Command	Remarks
Set index registers	800	*SD* 001 1	
	801	*SL* 004 1	
	802	*SB* 001 2	
	803	*SD* 001 2	
	804	*SL* 396 2	
	805	*SB* 001 1	
Command → *A*	806	*CA* 821 1	
Store it	807	*ST* 002 2	
Increment it	808	*AD* 825 1	
	809	*ST* 821 1	
Step second index-register	810	*II* 820 2	
	811	*CA* 830 0	Store special commands at beginning
	812	*ST* 000 0	and end of program
	813	*CA* 831 0	
	814	*ST* 001 9	
	815	*CA* 819 0	
	816	*ST* 399 0	
	817	*CA* 825 0	
	818	*ST* 002 0	
	819	*HA* 000 0	
Step first index-register	820	*II* 806 1	
	821	*TC* 805 0	
	822	*CA* 502 0	Constants
	823	*MU* 702 0	
	824	*AD* 499 0	
	825	*ST* 499 0	
	826	00 001 0	
	827	00 001 0	
	828	00 000 0	
	829	00 000 0	
	830	*CA* 501 0	
	831	*MU* 701 0	

very favorably. It is probably true that for some average class of problems the system in Sec. 17.23 is as good as the other methods. However, small differences in the logic of the computers as well as in the methods used may make one of the other methods more attractive.

Table 17.14. Comparison of Coding Methods

Discussed in Sec.	Name	No. of commands	No. of commands executed	Result
17.23	Cyclic code with index register	12	506	Answer
17.24	Cyclic code, no index register	18	1,203	Answer
17.25	Linear code	400	400	Answer
17.26	Generator	32	2,306	A linear code which will do the problem

17.28. INTERPRETIVE SYSTEMS

Almost all general-purpose digital computers have sufficient facilities that it is possible to establish command systems in the computer that make them operate like any other given digital computer (provided there is sufficient memory space). In actual practice interpretive systems are in limited use since they tend to make the computer slower.

In fixed-point binary computers an interpretive system may make the calculator look like a floating-decimal computer to the operator. Even in a decimal floating point machine, computer systems may be devised that double the precision or that make the operations handle complex numbers. In fact, a given interpretive system may be made to do operations on floating-point numbers, on complex numbers, on vectors, or on matrices, just by changing the system of subroutines.

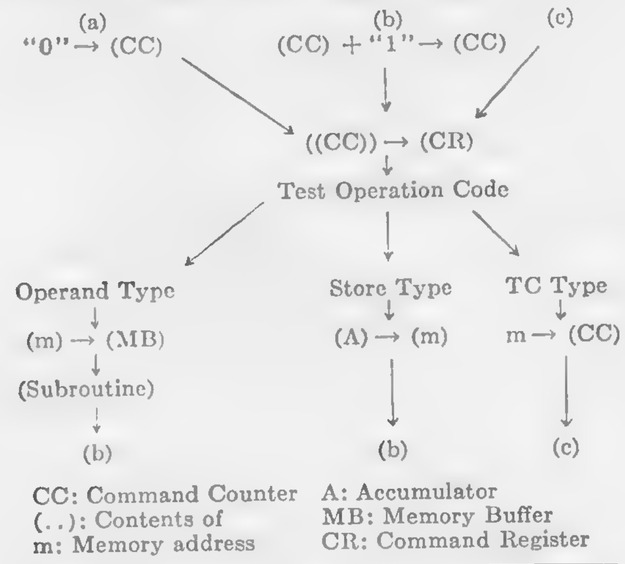


FIG. 17.5. Structure of an interpretive system.

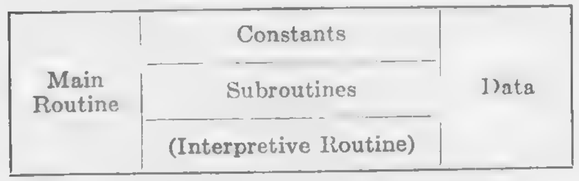


FIG. 17.6. Allocation of memory space.

Figure 17.5 shows the general structure of an interpretive system. There is a portion which imitates the command counter in a computer and which keeps track of where to obtain the pseudo commands from the memory of the computer. Another portion interprets the command and enters the proper subroutine. The subroutines all exit back to the command-counter portion.

Entry *a* sets the command counter, *CC*, to zero. The statement $(CC) \rightarrow (CR)$ means that the contents of the address specified by *CC* are transferred to the command register as the command (actually pseudo command) to be executed. All the commands of the pseudo code are classified into three groups. The first group is those that require an operand. This group includes all the arithmetic commands and the output commands. The second group is the store type and includes the store command and the input commands. Transfer-of-control commands (conditional, unconditional, mark transfers, return transfers, etc.) fall in the third group. In transfer

of control-type commands the address may replace the contents of the command counter, *CC*.

Figure 17.6 shows the general allocation of memory space. The interpretive routine includes the subroutines necessary to do the various arithmetic operations (floating decimal, for example), any input or output conversion, etc. The floating-point routine may need certain constants for command modification. All the rest of the memory serves as operating memory and may be used to store pseudo commands or data. Usually the data and commands may be intermixed with considerable freedom; so the diagram of Fig. 17.6 should be considered as functional.

The next section will consider a MARK TRANSFER approach to the problem of subroutine organization, which is an alternate method to the interpretive scheme described above.

17.29. A "MARK TRANSFER" APPROACH TO SUBROUTINE ORGANIZATION

In the interpretive system described in Sec. 17.28 an interpretive or executive routine controls the flow of the computation and the coding of the problem is done in a pseudo code, wherein each pseudo command can call into action subroutines which accomplish as elaborate sequences of operations as may be desired. For example, a

Table 17.15. Subroutine Characteristics

Routine	Entry	Operation
Floating addition	050	(099) $+ A \rightarrow$ (099) and A*
Floating subtraction	052	(099) $- A \rightarrow$ (099) and A
Floating multiplication	070	(099) $\cdot A \rightarrow$ (099) and A
Floating division	080	(099)$/A \rightarrow$ (099) and A

* A is the actual accumulator.

single command could accomplish the scalar product which was discussed in Secs. 17.23 to 17.26.

Suppose that the calculator operated in a fixed-point binary-number system, and an interpretive system had been prepared supplying subroutines which did the arithmetic operations in floating-point binary, and the input and output was in floating decimal. In this case the pseudo code might look very much like the code of the examples in Sec. 17.23 or 17.24 depending upon whether or not the system (not necessarily the calculator) had index registers. If scalar product was to be a pseudo command then more information would have to be supplied than the length of the commands in Sec. 17.23 or 17.24 provides. In such a command it would be necessary to designate the initial address of the x_i, and the initial address of the y_i, as well as where to place the answer, and it would be necessary to specify the number of components in the vectors. One solution would be to use the word following the command to supply this extra information, and to have the scalar-product routine itself step the command counter $(CC) + \text{"1"} \rightarrow (CC)$.

In this section it will be assumed that subroutines exist which accomplish floating-decimal arithmetic operations and that the computer has as actual machine commands both MARK PLACE and RETURN commands. Table 17.15 gives the characteristics of the subroutines and Table 17.16 shows the coding required to accomplish the computation. Each subroutine does a single floating-decimal operation such as $A + B = C$ where A, B, and C refer to fixed memory locations. In the illustrative example A has been assigned to be address 099 in the memory and B is to be the accumulator in the arithmetic unit of the calculator. C is both 099 and the accumulator. Each subroutine terminates with a return-transfer command which brings control back to the next following command in the main routine.

Table 17.16. Scalar Product Using Subroutines

Flow diagram	Location	Command	Remarks
$0 \rightarrow S$	500	*SB* 001 1	Set index register
$1 \rightarrow i$	501	*SD* 001 1	
	502	*SL* 100 1	
$0 \rightarrow S$	503	*CA* 098 0	$0 \rightarrow A$
	504	*ST* 100 0	$A \rightarrow S$
$S + x_i y_i \rightarrow S$	505	*CA* 100 1	
	506	*ST* 099 0	
	507	*CA* 300 1	
	508	*MT* 070 0	Multiply
	509	*CA* 100 0	
	510	*MT* 050 0	Add
	511	*ST* 100 0	
$i + 1 \rightarrow i$	512	*II* 505 0	
	513	*HA* 000 0	

17.30. COMMAND SYSTEMS

Computers have been constructed which have what is called one-address, two-address, three-address, and four-address command systems. Of these the three- and four-address systems have not been exploited very extensively commercially. In the one- and two-address systems results are left in the arithmetic register usually called the accumulator. In some cases another arithmetic register may be involved, for example, to store the multiplicand during the multiplication process. Generally, in the three- and four-address systems all operands are obtained from the memory and results are left in the memory.

The basic argument for three- and four-address systems was the statement that the operations specified by the commands were more nearly like the mathematical formulas involved in the computation than in the case of one- or two-address systems. Another basis of argument was the memory space required to specify the problem. This last issue was complicated by the question of whether the computer stored one or two commands in a given address.

It seems clear that there are going to be more built-in index registers in future computers, and this is a strong factor favoring single-address systems, since indexing in multiple-address systems is relatively complicated.

Single-address Systems. The list of commands of Sec. 17.17 is typical of a single-address computer except that very few computers have a square-root command or a negate command. On the other hand, most computers have a product command which produces the double-length answer for the product of two single-length numbers. This command and adequate overflow detection make it possible to do double precision arithmetic. In all single-address systems a command counter causes commands to be taken from consecutive locations in the memory.

Two-address System. Some computers have been developed in which the second address refers to an operand. However, the system which is most popular uses the second address, hereafter referred to as b, to specify the location of next command. The computers in which this system is used have a cyclic memory, and this use of the second address makes possible coding with minimum memory-access time. The first address is used in the same way as the address of the single-address system described above.

Three-address System. In the three-address system the first two addresses refer to the locations of the operands and the third is the location of the result. This

system, in effect, makes every memory location play the role of an accumulator. As in the single-address system the location of the next command is determined by a command counter. In three-address systems there is little use for clear and add, or clear and subtract, commands. If a number must be transferred into a particular location it may be necessary to store a 0 in some location so that the appropriate addition may be performed to accomplish the transfer.

Four-address System. Only a few computers have been built using four addresses in each command, and probably no more will be built. The reason for this is the extreme length of the command word in order to contain the four address for a reasonable size memory and, as was mentioned above, the difficulty in handling built-in indexing systems. The arithmetic operations here are similar to those of the three-address system. The fourth address provides for the location of the next command. In one such computer the next command is normally that next in sequence in the memory as determined by a command counter. However, in the case of overflow in addition and subtraction the fourth address determines the location of the next command. This allows the programmer to correct for overflow situations or to use this feature for branching in the program logic.

17.31. COMMANDS OF SPECIAL INTEREST

A number of commands appear in particular computers designed for special types of problems. Some are quite general in application while others are for particular fields. For example, some commands are particularly useful in business-applications problems. Usually the rest of the commands in the system are designed so as to work well in conjunction with these special commands.

17.31.1. Add-multiply Command. Consider a two-address system wherein the command is of the form *OP*, *a*, *b* where *OP* specifies the operation and the letters *a* and *b* specify two memory addresses. Conventional commands in such a system include addition, subtraction, and multiply commands similar to the single-address systems (cf. Sec. 17.18) where *b* may specify the location of next command; or the operations may be of the form $(a) + (A) \rightarrow (b)$, $(A) - (a) \rightarrow (b)$, $(A) \cdot (a) \rightarrow (b)$. In either case if the computer is to be used very extensively on vector- or matrix-type operations then an accumulate-multiply command is very useful. This would be of the form $(A) + (a) \cdot (b) \rightarrow (A)$. This permits operations such as scalar products with a minimum number of commands.

17.31.2. Repeat Command. Given a two-address system such as that described in the preceding section a repeat command is very powerful. For full flexibility it is convenient to have two more "address modifiers" as part of the command. Suppose that the command is of the form *OP*, *a*, *b*, *c*, *d*, where *OP*, *a*, and *b* are as described above. The *c* and *d* are modifiers of *a* and *b*, respectively, and consist of one digit each. If the repeat command is written as *RE*, a_r, b_r, c_r, d_r then the next following command is executed a_r times and with each operation the addresses *a* and *b* are increased by c_r and d_r, respectively. After the repetition is finished the location of the next command is specified by b_r.

With such a repeat command input and output commands can be relatively simple. For example, a magnetic-tape transfer command can transfer one word, with *a* specifying the high-speed memory location and *b* specifying the tape location. The repeat command would control how many words were transferred.

17.31.3. Shift-jump Command. In some problems it is required to choose the path at a branch in a certain fixed pattern (five times one way, three the other, two the first, etc.). If the total number of choices is less than the number of bits in a word and the computer is binary, such an operation can be specified by a jump-shift command. In a two-address system with negative numbers in complementary form the command causes a jump to location *a* if (*A*) is positive and to location *b* if (*A*) is negative, and the number in *A* is shifted left once. With negative numbers in complementary form the most significant digit is the sign, and left shifts can bring any pattern of 1's and 0's into that position, giving a corresponding pattern of jumps to locations *a* or *b*.

In a four-address computer this same operation has been mechanized as follows: Three of the addresses specify the operands and location of the result and the fourth is the location of the next command in case the operation overflows. If the three operand and result addresses are the same the number is doubled (equivalent to shifting left in the binary-number system) and stored back in the same location. Thus, with the same pattern of 1's and 0's as in the example above, the branching can be controlled in a similar fashion. The two possible paths are the location specified by the fourth address or the next in order.

17.31.4. Normalize. Many fixed-point decimal or binary computers have commands to facilitate programmed floating-point operation. An example of such a command is a normalize command which shifts a number left until there is a nonzero digit in the most significant position, and stores in some register the number of positions shifted.

A "position" command can do the inverse operation. That is, it will shift a number right an amount specified by the contents of another register. In applying this command in adding two floating-point numbers the exponents are subtracted, determining the amount that one of the operands is to be shifted right. Then the position command moves one of the sets of significant figures into position so that it can be directly added to the digits of the other operand.

17.31.5. Merge. In business applications items tend to have a relatively large number of digits, partly to identify the item and partly to furnish information about various aspects of it. For example, the identification may be a catalogue number and the item may also contain such information as the number on hand, the number on order, the value of each unit, reorder levels, and even such things as the manufacturer's or vendor's name and address. The sets of such items are called files and may be stored on punched cards or magnetic tape. When new items are to be added to the file they must be placed in the proper position, perhaps in order with respect to their identification number or code. This process of assembling two files to obtain a single file ordered in the same way is called *merging*.

A computer has been designed in which this process is accomplished by a single command provided that all the required information is in the high-speed memory and there is space for the resulting file. If there is not space then a single command will carry on the process until one of the two input files is exhausted or the resulting merged file has filled the allotted space, and will specify one of three places as the location of the next command, depending upon which of the three possibilities happened. This makes it very easy to program the merging of large files of data partially stored on magnetic tape.

17.31.6. Sift. Another example of a command which has applications in business-type problems and in scientific computing involves table look-up. The sift command counts in a list the number of items which have their code less than a given code, and finds the location of the first item whose code exceeds the given code.

17.31.7. Summarize. This command accumulates algebraically the contents of specified fields contained in successive multiple-word items stored in the memory. This command and the sift command described above can be used to provide a summarization of all those items whose control keys (identification) are identical. This is very useful in summary-report preparation.

17.32. COMPILERS

In Secs. 17.1 to 17.9 a general language of programming was discussed. In Secs. 17.10 to 17.31 details of various kinds of command systems and the process of coding were studied. In the past there has been no well-defined language in which to express the details of a computation, and the process of producing the commands for a problem on a computer was a manual process. This is changing in both respects since an algorithmic language is being developed[29] and compilers (computer programs to produce problem codes) are being developed which will accept such an algorithmic language and produce computer codes for particular computers.

Various compilers each with some of the features which are required in a complete algorithmic system have been developed and tried over the last few years. These

have been sufficiently successful so that many persons are looking forward to a solution of their coding problems (not programming problems) in terms of automatic techniques.

17.33. AN INTRODUCTION TO ALGOL 60*

This is an account of the more important features of the algorithmic language ALGOL 60. The aim is to explain rather than to define the language, and for full details reference should be made to the Report published in *Numerische Math.*, **2**, 106–136 (1960) or in *Comm. Assoc. Comp. Mach.*, **3**, 299–314, May, 1960. This description of ALGOL was first published in *The Computer Journal*, vol. 3, no. 2, in July, 1960 and is republished here by permission of the honorary editors. This article, Sec. 17.33, is published with the permission of the Director of the National Physical Laboratory.

17.33.1. Introduction. *Purpose.* ALGOL 60 is a formal symbolic language for expressing processes of computation ("algorithms").

The order code for any individual computer is in fact such a language—it has a "syntax" or set of rules specifying what are meaningful combinations of symbols, and a "semantics" or set of rules specifying the meanings of these combinations, that is to say the action taken by the computer when the orders are executed.

Such "machine languages" are burdened with details of the particular structure, arithmetic facilities, input-output and storage arrangements of the computer concerned. The purpose of ALGOL 60 is to provide a language for the expression of computation processes to the extent that they are independent of these details, and in a form which can be translated automatically into any particular machine code by a suitable translator program written for the computer concerned. The "machine oriented" details of the computation are to be embodied in the translator rather than in the ALGOL 60 program.

A second use of ALGOL 60 is for expressing processes of computation in a form suitable for human consumption. Whereas an unbroken sequence of symbols of a few different sorts is satisfactory for input to a computer, legibility requires a display on the printed page and a variety of forms of expression which will reflect the natural subdivision of the process. To accommodate these varying requirements there are three forms of ALGOL 60. It is defined in a form known as the "reference language" (employing a total of 116 basic symbols but in a linear fashion), used as input to translators in an appropriate "hardware representation" (employing, e.g., 5-bit characters on punched paper tape), and used in publications in the form of "publication language." The latter admits the use of suffixes ($a_{i,j}$ corresponding to $a[i,j]$ in the reference language), exponents (a^3 corresponding to $a\uparrow 3$), Greek letters, spaces, and line groupings of characters. Apart from such relatively trivial differences the three forms of ALGOL 60 are identical in content.

Words printed in bold type such as **begin** represent single basic symbols. In typescript these would be underlined.

Subject Matter. The computing processes described deal principally with ordinary real numbers, integers, and arrays of these. The real numbers are of necessity approximated digitally, but how this is done is not expressed.

In order to avoid reference to particular storage locations a notation is introduced for variable quantities in store, and it is important to be clear as to its precise significance. A letter, or a string of letters or decimal digits of which the first is a letter, is called an *identifier*. An identifier may be used in an ALGOL 60 program as a *simple variable*. This means that the program when ultimately translated and run on a computer will associate a particular storage location with that identifier. The number held in this store at any stage in the calculation is called the "current value" of the variable. An instruction or "statement" in the ALGOL 60 program which contains this identifier calls either for the use of this number in calculation or for its replacement by the result of calculation—the latter is referred to as "assignment of a value to the variable."

* By Michael Woodger.

Calculation Rules. The course of the computation process described by the ALGOL 60 program may be visualized as a succession of assignments of values to variables. The program expresses rules for the calculation of these assigned values, and these rules take a variety of forms. The simplest rules are provided by ordinary algebraic expressions compounded from simple variables and numerical (decimal) constants by the usual symbols for addition, subtraction, multiplication, and division. If E denotes such an expression the instruction in ALGOL 60 which calls for the assignment of the current value of E to a variable V is written

$$V := E$$

E may well contain V itself; for example the assignment statement $V := V + 1$ increases the current value of V by *1*.

To indicate that the variable V is to be treated as a real variable (say), rather than as an integer or other "type" of quantity, a *type declaration*

real V

is used. A succession of type declarations for the variables employed, followed by a succession of assignment statements, each statement and declaration being separated from its neighbours by a semicolon (;), and the whole being enclosed between the "statement brackets" **begin** and **end**, would constitute a simple ALGOL 60 program.

Arrays. Large blocks of numerical data are commonly treated as *arrays*, the elements of which are specified by sets of integers. The number of integers in such a set is the "dimensionality" of the array. Thus a matrix A is a two-dimensional array for which a set of two integers, the row number i and column number j, is used to single out a particular element. This element is denoted by the suffixed symbol A_{ij}. If the elements of a matrix A of n columns are stored in consecutive locations row by row the position number of element A_{ij} relative to A_{11} as number 0 is $(i - 1)n + (j - 1)$, and this rule for finding the (i,j) element must be available to the computer whenever the value of this element is called for.

An array is in effect a function of a number of integer variables, the values of the function being explicitly listed (stored) and the values of the variables being used to locate the function value in the list (store). This may be contrasted with the situation with an algebraic expression, which is effectively a function of the variables it contains. In that case no function values are stored but the values of the variables are used to calculate the function value by a formula. A formula can generally be used for arbitrary values of the arguments. An array on the other hand is limited, for reasons of storage, in each of its dimensions. ALGOL 60 deals only with arrays whose dimensions are independent, i.e., generalized rectangular arrays, and in particular only those employing consecutive integer suffix values. The extent of an array is specified by the lower and upper limiting values of each of its suffixes and this information for each array, together with the type of its elements, is provided by an *array declaration.*

Each element of an array may be assigned a value independently of the others. The general form of such an assignment statement is

$$V\,[E, E, \ldots, E] := F$$

or in "publication language" ALGOL 60 (which permits the use of suffixes instead of square brackets)

$$V_{E,E,\ldots,E} := F$$

Here V is a *subscripted variable* (the identifier naming the array), each E (a *subscript*) is an expression whose current value is used to locate the appropriate element of the array, and F is the expression whose current value is to be assigned to that element.

Constituents of a Program. The basic constituents of an ALGOL 60 program are thus *statements* which are executed as instructions in the order in which they are written and have the effect of assigning values to certain variables, and *declarations* which are not themselves executed as instructions but provide information necessary to the execution of the statements following them.

There is a third (unnamed) category of constituents which corresponds to the control or "red tape" instructions in machine language. These make it possible

1. To break off a sequence of calculations and start again somewhere else in the program (the "jump" or *go to* statements)
2. To skip certain statements in a sequence if certain conditions are not satisfied (*conditional* statements), and
3. To repeat the next statement for a succession of values of a variable (the *for* statements)

Since no reference is made to storage locations of the ALGOL 60 statements themselves they have to be *labeled* to be identified as destinations for *go to* statements. Any identifier or unsigned integer may be used as a label, written in front of the statement labeled and separated from it by a colon (:).

Sequences of statements may be combined within the statement brackets **begin** and **end** to form *compound statements*, which again may be labeled.

Blocks and Declarations. Each identifier used in an ALGOL 60 program, other than as a label, is introduced by a declaration which gives information concerning it and which is referred to by the translator when executing the statements in which the identifier appears. This declaration is written (possibly with others) following the **begin** symbol of some compound statement (possibly the whole program). A compound statement containing declarations in this way is called a *block*, and each declaration is valid only for the block to which it is attached. This means that each identifier I is "local" to the block B for which it is declared, in the sense that on exit from B either via the **end** (on completion of the last statement in B) or on execution of a *go to* (jump) statement leading outside B, I has no longer the declared significance and may be used afresh in a new declaration to denote some entirely different thing. Alternatively, if the block B is itself a component statement of a larger block A for which I was already declared, then while B is being executed I has the local significance declared for B but on exit to A it reverts to the significance it had when B was entered, its value (if a variable) remaining unaffected by passage through B. Identifiers used in A and not declared for B retain their significance within B. Thus every block automatically introduces a new level of nomenclature.

Labels are automatically "local" to the blocks in which they are used, so that a jump into a block from outside is not possible—that would bypass the governing declarations.

A declaration D, attached to a block B and governing an identifier I representing a variable or array, may be prefixed with the symbol **own**. This has the effect that on reentering B the value of the variable (or values of the array elements) is as it was left at the previous exit from B.

Functions and Procedures. We have noted that an algebraic expression can be considered as a rule for evaluating a function, the constituent variables in the expression being the arguments of the function. More generally, the same is true of any ALGOL 60 program if we single out one of the variables to which the program assigns a value as being the value of the function. ALGOL 60 provides for the definition of a function in this way, giving it a name and indicating the variables concerned, and for using its name (followed by a parenthesized list of expressions to be used as its arguments) as a constituent of algebraic expressions elsewhere, i.e., as a "function designator." A notation is also provided for defining as a "procedure" any ALGOL 60 program even when its effect is not simply the assignment of a value to a single variable. In this case the procedure name (followed by a parenthesized argument list as before) appears not as part of an algebraic expression but as a statement—a "procedure statement"—in the program. Both functions and procedures are introduced and defined by *procedure declarations* which comprise the defining block of program or statement (the "procedure body") prefixed by the symbol **procedure**, the name of the procedure (any identifier), and details regarding which identifiers in it are to be treated as the arguments or "parameters" and how they are to be used when the procedure is called. A type declaration in front of the symbol *procedure* indicates that a function is being defined and gives the type of its value.

For example the procedure declaration

real procedure *sumsq* (P, Q, R, S); *sumsq* $:= P^2 + Q^2 + R^2 + S^2$

might be used to define the function designator appearing in a statement such as

$$y := T^2 + sumsq\ (a - m, b - m, c - m, d - m)/3$$

When this statement is executed the effect is as if sumsq represented a simple variable to which a value was first assigned by *substituting* the "actual parameters" $a - m$, $b - m$, $c - m$, $d - m$ respectively for the "formal parameters" P, Q, R, S in the procedure body and then executing the resulting assignment statement. $T^2 + sumsq/3$ is then calculated and its value assigned to y.

It should be noted that the letters P, Q, R, S (in general any identifiers) bear no relation to other identifiers outside the procedure body and do not represent "variables"—they are used simply to mark positions for substitution in the procedure body and for this reason are called formal parameters.

A simple example of a procedure statement is

$$TEST\ (b^2 - 4 \times a \times c, L1, L2, L3)$$

which might be used as a three-way discrimination using current values of variables a, b, c to continue the program at statements labeled $L1$, $L2$, or $L3$ according as $b^2 - 4 \times a \times c$ is positive zero or negative. The procedure declaration defining $TEST$ (which must appear in the head of a block in which the above statement appears) could be as follows:

procedure $TEST$ (a, P, Z, N); **if** $a > 0$ **then go to** P **else if** $a = 0$ **then go to** Z **else go to** N

This illustrates also a simple form of conditional statement. If B represents some condition which may or may not be currently satisfied and S, T represent statements the effect of

if B **then** S **else** T

is the same as that of S if B holds and is the same as that of T if B does not hold. The statement

if B **then** S

has no effect (is skipped) if B does not hold.

Standard Functions. It is recommended in the Report that the following identifiers should be reserved for standard functions which may be available with a particular translator without explicit declaration, through the use of a library of subroutines. x denotes the current value of the expression E.

abs (E)	the modulus (absolute value) of x
sign (E)	$+1$ if $x > 0$; 0 if $x = 0$; -1 if $x < 0$ } of type **integer**
entier (E)	the integral part of x (largest integer not greater than x), of type **integer**
sqrt (E)	square root of x
sin (E)	sine of x
cos (E)	cosine of x
arctan (E)	principal value of arctangent of x
ln (E)	natural logarithm of x
exp (E)	exponential function of x

Input and Output. The input of data and the output of results have been excluded from detailed consideration in the reference language but it is intended that suitable procedures be available with individual translators which will perform these functions. These procedures would be referred to by name in the ALGOL program and might require as parameters information concerning the layout of data on the external medium, i.e., information not expressible in the language. For this purpose arbitrary strings of ALGOL 60 symbols may be entered as actual parameters of procedures through the use of two special "string quotes" symbols, and the body of a procedure declaration may be expressed in machine code while the heading remains expressed in ALGOL 60 for reference.

17.33.2. Examples. Before indicating the full generality of the various categories of symbolism provided in ALGOL 60 we give first some elementary examples.

The following procedure declaration defines *max* (*a*, *n*) as the function whose arguments are a vector (one-dimensional array) *a* of *n* elements, and the integer *n*, and whose value is the modulus of the largest component of the vector.

```
real procedure max (a, n);
L1: begin real m; m := 0; L2: for i := 1 step 1 until n do
L3: begin real x; x := a_i; if x < 0 then x := −x; if x > m then m := x end;
L4: max := m end
```

This illustrates a simple kind of "for" statement—the statement labeled *L2* causes the statement *L3* to be executed *n* times with *i* taking the values *1*, *2*, . . . , *n* in turn before proceeding to statement *L4*. The labels are superfluous in this case since no *go to* statement is used.

It is worth noting that one cannot eliminate *m* and write the procedure body simply as

```
L1: begin max := 0; L2: for i := 1 step 1 until n do
L3: begin real x; x := a_i; if x < 0 then x := −x; if x > max then max := x end end
```

because wherever the identifier *max* appears other than on the left of an assignment statement—in this case in the condition ($x > max$)—it calls into use the procedure *max* itself, which is not the intention here.

To illustrate the use of the above procedure declaration by a function designator in an algebraic expression, suppose we have a matrix *B* of *r* rows and *c* columns and we wish to normalize its rows to have largest elements unity (where the rows have at least one nonzero element). We could write:

```
    for i := 1 step 1 until r do
1:  begin array a [1 : c]; for j := 1 step 1 until c do a_j := B_i,j;
    for j := 1 step 1 until c do
    if max (a, c) ≠ 0 then B_i,j := B_i,j/max (a, c) end
```

Here an array *a* local to the block labeled *1* has been introduced and a copy of the current *ith* row of *B* assigned to it simply in order to have a one-dimensional array to use as actual parameter of *max*. *a* is defined by the array declaration **array** *a* [*1* : *c*] as being one dimensional with suffix values ranging from *1* to *c*, and because no "type" is indicated the type **real** is understood (a special convention for array declarations). *B* could have been introduced by a declaration of the form

```
array B [1 : r, 1 : c]
```

The following example from the ALGOL 60 report illustrates a more general form of the heading of a procedure declaration.

```
procedure Innerproduct(a, b)order:(k, p)Result:(y);
value k;
integer k, p; real y, a, b;
begin real s; s := 0;
for p := 1 step 1 until k do s := s + a × b;
y := s end Innerproduct
```

The formal parameters of this procedure are *a*, *b*, *k*, *p*, *y* and could have simply been listed between a single pair of parentheses in the first line. The convention used here is that any separating comma ("parameter delimiter") may be replaced without effect on the program by a string of letters followed by a colon and enclosed between reversed parentheses thus

```
)order:(
```

By this means the various parameters may more easily be recognized by the reader.

The last line illustrates a similar convention that the symbol **end** may be followed, without effect, by any sequence of symbols not containing **end** or **else** or a semicolon.

The second line of the declaration is known as the "value part," and has been absent from the examples given earlier. It indicates those parameters which are to be "called by value" rather than "called by name" as hitherto. This means that

at a procedure call these formal parameters in the procedure body are *not* to be replaced by the actual parameters but are to be treated in the execution of the procedure as if they were identifiers local to the procedure body, representing variables or arrays to which were initially assigned the current values of the actual parameter expressions.

The third line of the declaration is called the "specification part" and provides in general information concerning the kinds and types of admissible actual parameters.

In this example the "running variable" *p* of the "for" statement has deliberately been made one of the parameters so that it may also appear as a suffix in the actual parameter expressions substituted for *a* and *b*. Thus a particular procedure statement employing this procedure to form the inner product of a vector *B* of order *10* and a vector defined by fixing the first and third suffix of a three-dimensional array *A* might be

```
Innerproduct (A [t, P, u], B [P], 10, P, Y)
```

Of course the procedure could also be abused (presumably successfully) by writing *Innerproduct*(*C*, *D*, *n*, *i*, *E*) to have the same effect as

```
E := n × C × D
```

in which *C*, *D*, and *n* are simple variables, the last a positive integer.

By omitting the parameter *y*, replacing the last assignment statement by *Innerproduct* := *s*, and preceding the symbol **procedure** by **real** this procedure declaration would be changed into a definition of *Innerproduct* as a function designator.

17.33.3. Syntax. In order to describe precisely what combinations of symbols are meaningful in ALGOL 60 a special notation is used which is best explained by an example. Given that the symbol ⟨letter⟩ names any one of the fifty-two lower or upper case Latin letters, and that ⟨digit⟩ names any one of *0*, *1*, . . . , *9*, we can define the name ⟨identifier⟩ by the formula

```
⟨identifier⟩ : : = ⟨letter⟩|⟨identifier⟩⟨letter⟩|⟨identifier⟩⟨digit⟩
```

in which the vertical strokes may be read "or." This is to be read as saying that an identifier is either a letter or is an identifier followed by a letter or is an identifier followed by a digit. It defines the notion of identifier recursively. Thus *V23a* is an identifier because *V23* is one, *V23* because *V2* is, and *V2* because the single letter *V* is. This corresponds with the verbal definition under 17.33.1 above as "a letter or a string of letters or decimal digits of which the first is a letter."

In general, the juxtaposition of names in these "metalinguistic formulae" is used to name the juxtaposition of the sequences of symbols named, and any ALGOL 60 symbol appearing is used to name itself. Thus examples of sequences named by +⟨digit⟩ · ⟨digit⟩ would be *+3.7*, *+0.0*, *+9.9*.

In the next section the full extent of the more important means of expression in ALGOL 60 will become apparent.

17.33.4. Syntax of Expressions. *Numbers.* Constant numbers are expressed in ALGOL 60 in the decimal notation with sign, integral part, decimal point, fractional part, and finally a signed power of 10 to be used as a scale factor. The exponent 10 is brought down to the level of the text by using a special suffix symbol ₁₀ for the radix, and this and various parts of the general form may be omitted when superfluous, e.g., *−1.083₁₀−02*, *.7300*, *+₁₀7*. The precise definition in several steps is as follows.

```
⟨unsigned integer⟩ : : = ⟨digit⟩|⟨unsigned integer⟩⟨digit⟩
⟨integer⟩ : : = ⟨unsigned integer⟩|+⟨unsigned integer⟩|−⟨unsigned integer⟩
⟨decimal fraction⟩ : : = ·⟨unsigned integer⟩
⟨exponent part⟩ : : = ₁₀⟨integer⟩
⟨decimal number⟩ : : = ⟨unsigned integer⟩|⟨decimal fraction⟩|
                        ⟨unsigned integer⟩⟨decimal fraction⟩
⟨unsigned number⟩ : : = ⟨decimal number⟩|⟨exponent part⟩|
                        ⟨decimal number⟩⟨exponent part⟩
 number⟩ : : = ⟨unsigned number⟩|+⟨unsigned number⟩|−⟨unsigned number⟩
```

Variables

⟨simple variable⟩ : : = ⟨identifier⟩
⟨array identifier⟩ : : = ⟨identifier⟩

Although strictly speaking redundant these definitions help to convey part of the meaning at the same time as the structure is defined.

⟨subscript list⟩ : : = ⟨arithmetic expression⟩|⟨subscript list⟩,⟨arithmetic expression⟩
⟨subscripted variable⟩ : : = ⟨array identifier⟩[⟨subscript list⟩]
⟨variable⟩ : : = ⟨simple variable⟩|⟨subscripted variable⟩

Function Designators

⟨actual parameter⟩ : : = ⟨string⟩|⟨expression⟩|⟨array identifier⟩|⟨switch identifier⟩|⟨procedure identifier⟩

Strings are not discussed in this article. Switches are defined under *Switches, Designational Expressions.*

⟨letter string⟩ : : = ⟨letter⟩|⟨letter string⟩⟨letter⟩
⟨parameter delimiter⟩ : : = ,|)⟨letter string⟩:(
⟨actual parameter list⟩ : : = ⟨actual parameter⟩|⟨actual parameter list⟩⟨parameter delimiter⟩⟨actual parameter⟩
⟨procedure identifier⟩ : : = ⟨identifier⟩
⟨function designator⟩ : : = ⟨procedure identifier⟩|⟨procedure identifier⟩(⟨actual parameter list⟩)

Simple Arithmetic Expressions

⟨adding operator⟩ : : = +|−
⟨multiplying operator⟩ : : = ×|/|÷
⟨primary⟩ : : = ⟨unsigned number⟩|⟨variable⟩|⟨function designator⟩|(⟨arithmetic expression⟩)
⟨factor⟩ : : = ⟨primary⟩|⟨factor⟩↑⟨primary⟩
⟨term⟩ : : = ⟨factor⟩|⟨term⟩⟨multiplying operator⟩⟨factor⟩
⟨simple arithmetic expression⟩ : : = ⟨term⟩|⟨adding operator⟩⟨term⟩|⟨simple arithmetic expression⟩⟨adding operator⟩⟨term⟩

The operators +, −, × have the conventional meaning and yield expressions of type **integer** when both operands have type **integer,** otherwise **real.**

The operator / denotes division, to be understood as a multiplication of the preceding term by the reciprocal of the following factor, the result having always type **real.**

The operator ÷ also denotes division, but is restricted to operands of type **integer** and yields the integral part of the quotient, defined by

$$m \div n = sign\ (m/n) \times entier\ (abs\ (m/n))$$

The operator ↑ denotes exponentiation, the preceding factor being the base and the following "primary" the exponent. If the exponent is a positive integer i the result is a product of i equal factors of the same type as the base, if a negative integer then the result is the reciprocal of such a product and of type **real,** or is undefined if the base is zero, while if the exponent is zero the result is unity of the same type as the base. If the exponent r is of type **real** and the base a is positive the result is $exp\ (r \times ln\ (a))$ of type **real;** if a is zero the result is zero of type **real** for positive r, and otherwise is undefined.

The sequence of operations in evaluating an expression is from left to right subject to the usual use of parentheses and to the following rules of precedence between the operators, which are reflected in the syntax above:

first: ↑
second: × / ÷
third: + −

Boolean Expressions. These are used chiefly to express conditions for use in conditional statements and, as introduced below, in conditional expressions. For these purposes a condition is essentially a two-valued thing, it either holds or does not hold, and one can manipulate conditions as two-valued variables by the algebra of logic, i.e., Boolean algebra, which is available as part of ALGOL 60. A variable may be declared to have type **Boolean,** and its value is then either **true** or **false** (these being basic symbols). Such a variable may be assigned the current value of a Boolean expression just as in ordinary arithmetic. A function designator may have type **Boolean,** and one can have a Boolean array.

The simplest conditions are relations of equality or inequality between arithmetic expressions. From any condition we can derive the reverse condition by preceding it by the logical negation sign ¬, and from any two conditions we can derive by the logical operators ∧ (and), ∨ (or), ⊃ (implies) and ≡ (equivalent) compound conditions expressing respectively that both hold, that at least one holds, that either the first does not hold or the second does hold or both hold, and that they either both hold or both do not hold.

For example

if $(a > b + 1) \vee (a < b - 1)$ **then go to** L

This is equivalent to

if $\neg\ abs(a - b) \leq 1$ **then go to** L

and also to

$B : = abs(a - b) \leq 1$; **if** $\neg\ B$ **then go to** L

which illustrates the "calculation" of a condition.

The sequence of operations in evaluating a Boolean expression follows the rules for arithmetic expressions, with the rules of precedence extended thus:

4th: $< \leq = \geq > \neq$
5th: ¬
6th: ∧
7th: ∨
8th: ⊃
9th: ≡

so the condition in the first example above could be written without parentheses as $a > b + 1 \vee a < b - 1$.

The syntax built up following this order of precedence is as follows:

⟨relational operator⟩ : : = <|≤|=|≥|>|≠
⟨relation⟩ : : = ⟨arithmetic expression⟩⟨relational operator⟩⟨arithmetic expression⟩
⟨logical value⟩ : : = **true**|**false**
⟨Boolean primary⟩ : : = ⟨logical value⟩|⟨variable⟩|⟨function designator⟩|⟨relation⟩|(⟨Boolean expression⟩)
⟨Boolean secondary⟩ : : = ⟨Boolean primary⟩|¬ ⟨Boolean primary⟩
⟨Boolean factor⟩ : : = ⟨Boolean secondary⟩|⟨Boolean factor⟩ ∧ ⟨Boolean secondary⟩
⟨Boolean term⟩ : : = ⟨Boolean factor⟩|⟨Boolean term⟩ ∨ ⟨Boolean factor⟩
⟨implication⟩ : : = ⟨Boolean term⟩|⟨implication⟩ ⊃ ⟨Boolean term⟩
⟨simple Boolean⟩ : : = ⟨implication⟩|⟨simple Boolean⟩ ≡ ⟨implication⟩

Conditional Expressions. These provide a method (due to J. McCarthy) for selecting one of a sequence of expressions $E_1, E_2, \ldots$ according to the current values of a corresponding sequence of conditions (Boolean expressions) $B_1, B_2, \ldots$. If B_k is the first condition to hold, i.e., to have the value **true,** then the value of the following conditional arithmetic expression is that of E_k, regardless of whether or not $B_{k+1}, B_{k+2}, \ldots$ or $E_{k+1}, E_{k+2}, \ldots$ are even defined.

if B_1 **then** E_1 **else if** B_2 **then** E_2 **else** $\ldots$ **else** E_n

In particular $E_1, E_2, \ldots$ may themselves be Boolean expressions.

This enables us to complete the syntax of arithmetic and Boolean expressions as follows:

⟨if clause⟩ : : = **if** ⟨Boolean expression⟩ **then**
⟨Boolean expression⟩ : : = ⟨simple Boolean⟩|
⟨if clause⟩⟨simple Boolean⟩**else**⟨Boolean expression⟩
⟨arithmetic expression⟩ : : = ⟨simple arithmetic expression⟩|
⟨if clause⟩⟨simple arithmetic expression⟩**else**⟨arithmetic expression⟩

Discontinuous functions are naturally expressed as conditional expressions. For example

sign (*E*) is equivalent to **if** $E > 0$ **then** *1* **else if** $E = 0$ **then** *0* **else** -1
and *abs* (*E*) is equivalent to **if** $E < 0$ **then** $-E$ **else** E

Switches, Designational Expressions. Just as an arithmetic expression is a rule for obtaining a numerical value, so a designational expression is a rule for obtaining a designation of a statement, i.e., a label. The only such expression so far introduced is simply a label itself, but we now provide a notation—the "switch"—for using any arithmetic expression whose value is a positive integer to select one from a list of labels (or again in general designational expressions). This is in some ways analogous to a one-dimensional array. Whereas an array declaration merely specifies size, shape, and type of values, a *switch declaration* actually exhibits the list of values. The designational expression whose value is the kth label in this list is written like a suffixed variable with k as suffix and the switch identifier as identifier.

For example if the switch S were defined by the declaration

switch $S := L, P, 4, L$

then the "switch designators" $S[1]$ and $S[4]$ and $S[3 + 1\uparrow 1]$ have each as value the label L. These may themselves appear in *go to* statements within the block to which the switch declaration is attached, or as entries in other switch declarations.

Finally one can have conditional designational expressions just as in the case of arithmetic and Boolean expressions, and the full syntax is as follows.

⟨label⟩ : : = ⟨identifier⟩|⟨unsigned integer⟩
⟨switch identifier⟩ : : = ⟨identifier⟩
⟨switch designator⟩ : : = ⟨switch identifier⟩[⟨arithmetic expression⟩]
⟨simple designational expression⟩ : : = ⟨label⟩|⟨switch designator⟩|
(⟨designational expression⟩)
⟨designational expression⟩ : : = ⟨simple designational expression⟩|
⟨if clause⟩⟨simple designational expression⟩**else**⟨designational expression⟩
⟨expression⟩ : : = ⟨arithmetic expression⟩|⟨Boolean expression⟩|
⟨designational expression⟩

17.33.5. Syntax of Statements. *Assignment Statements.* The only new point here is the provision for simultaneous assignments such as $a[b[1,1]] := x := b[1,1] := 3 \times x$ which assigns the current value of $3 \times x$ to a certain element of the one-dimensional array a (depending on the current value of $b_{1,1}$), to x itself, and then to $b_{1,1}$. (All variables here must be integers.)

If the expression assigned is E of type **real**, the variables may be of type **integer** and then the value assigned to them is that of *entier*$(E + 0.5)$.

⟨left part⟩ : : = ⟨variable⟩ : =
⟨left part list⟩ : : = ⟨left part⟩|⟨left part list⟩⟨left part⟩
⟨assignment statement⟩ : : = ⟨left part list⟩⟨arithmetic expression⟩|
⟨left part list⟩⟨Boolean expression⟩

Go to Statements

⟨go to statement⟩ : : = **go to**⟨designational expression⟩

Dummy Statements

⟨dummy statement⟩ : : = ⟨empty⟩
⟨empty⟩ names the 'empty' sequence of symbols.

A dummy statement is simply a blank—nothing is written. Its only purpose is to place a label. A *go to* statement which is simply to lead out of a compound statement to the following one can refer to a labeled dummy statement before the **end,** thus:

begin . . . ; **if** $x < 0$ **then go to** *END;* . . . ; *END:* **end**

Procedure Statements. These are syntactically identical with function designators (17.33.4 above) but are used as statements instead of as expressions.

⟨procedure statement⟩ : : = ⟨procedure identifier⟩|
⟨procedure identifier⟩(⟨actual parameter list⟩)

The actual parameter list might be absent, e.g., in the case of a procedure which simply sounded an alarm; the procedure itself would then be in machine code.

For Statements. A "for clause" causes the statement S which it precedes to be executed zero or more times, and performs an assignment of a value to a "controlled variable" V immediately before each execution. The "for list" defines these consecutively assigned values. An element of this list which is an arithmetic expression causes the assignment of its current value to V and a single execution of S. An element of the form A**step**B**until**C where A, B, C are arithmetic expressions causes the assignment of the current values of A, $A + B$, $A + 2B$, . . . to V and corresponding execution of S, the operation terminating as soon as $V - C$ has the same sign as B (this test being made just prior to each execution, so that S is never executed if initially $A - C$ and B have equal signs. An element of the form E**while**F where E is an arithmetic expression and F a Boolean expression causes the assignment of the current value of E prior to each execution, this being repeated until the current value of F is **false,** the test being made prior to each execution as above. This additional facility enables the number of times a loop is executed to be made to depend on the results of calculation without the necessity of writing a special test instruction.

If S is left by a *go to* statement, interrupting the execution of the *for* statement, the then current value of V continues to be available outside; otherwise V is treated as local to the *for* statement.

⟨for list element⟩ : : = ⟨arithmetic expression⟩|
⟨arithmetic expression⟩**step**⟨arithmetic expression⟩**until**⟨arithmetic expression⟩|
⟨arithmetic expression⟩**while**⟨Boolean expression⟩
⟨for list⟩ : : = ⟨for list element⟩|⟨for list⟩,⟨for list element⟩
⟨for clause⟩ : : = **for**⟨variable⟩ : = ⟨for list⟩**do**
⟨for statement⟩ : : = ⟨for clause⟩⟨statement⟩|⟨label⟩:⟨for statement⟩

The last definition shows that *for* statements may be repeatedly labeled. This is true of all kinds of statement, as will be seen.

Conditional Statements, Compound Statements, Blocks. The following completes the syntax of statements:

⟨basic statement⟩ : : = ⟨assignment statement⟩|⟨go to statement⟩|
⟨dummy statement⟩|⟨procedure statement⟩|⟨label⟩:⟨basic statement⟩
⟨unconditional statement⟩ : : = ⟨basic statement⟩|⟨for statement⟩|
⟨compound statement⟩|⟨block⟩
⟨if statement⟩ : : = ⟨if clause⟩⟨unconditional statement⟩|⟨label⟩:⟨if statement⟩
⟨conditional statement⟩ : : = ⟨if statement⟩|⟨if statement⟩**else**⟨statement⟩
⟨statement⟩ : : = ⟨unconditional statement⟩|⟨conditional statement⟩
⟨compound tail⟩ : : = ⟨statement⟩**end**|⟨statement⟩;⟨compound tail⟩
⟨compound statement⟩ : : = **begin**⟨compound tail⟩|⟨label⟩:⟨compound statement⟩
⟨block head⟩ : : = **begin**⟨declaration⟩|⟨block head⟩;⟨declaration⟩
⟨block⟩ : : = ⟨block head⟩;⟨compound tail⟩|⟨label⟩:⟨block⟩

17.33.6. Syntax of Declarations

⟨declaration⟩ : : = ⟨type declaration⟩|⟨array declaration⟩|⟨switch declaration⟩|
⟨procedure declaration⟩

Type Declarations

⟨type list⟩ : : = ⟨simple variable⟩|⟨simple variable⟩,⟨type list⟩
⟨type⟩ : : = **real**|**integer**|**Boolean**
⟨local or own type⟩ : : = ⟨type⟩|**own**⟨type⟩
⟨type declaration⟩ : : = ⟨local or own type⟩⟨type list⟩

Array Declarations

⟨lower bound⟩ : : = ⟨arithmetic expression⟩
⟨upper bound⟩ : : = ⟨arithmetic expression⟩
⟨bound pair⟩ : : = ⟨lower bound⟩:⟨upper bound⟩
⟨bound pair list⟩ : : = ⟨bound pair⟩|⟨bound pair list⟩,⟨bound pair⟩
⟨array segment⟩ : : = ⟨array identifier⟩[⟨bound pair list⟩]|
⟨array identifier⟩,⟨array segment⟩
⟨array list⟩ : : = ⟨array segment⟩|⟨array list⟩,⟨array segment⟩
⟨array declaration⟩ : : = **array**⟨array list⟩|⟨local or own type⟩**array**⟨array list⟩
Examples

array*a,b,c*[1:*n*,2:*m*],*s,u*[−2 × *r*:10]

declares three matrices *a*, *b*, *c* and two vectors *s* and *u*.

own Boolean array *peter3*[1 + (**if** *n* ≥ 0 **then** *n* **else** 0):20]

declares a vector called *peter3* with elements each either **true** or **false**, the definition of the lower bound of its suffix involving a conditional arithmetic expression.

Switch Declarations

⟨switch list⟩ : : = ⟨designational expression⟩|⟨switch list⟩,⟨designational expression⟩
⟨switch declaration⟩ : : = **switch**⟨switch identifier⟩ : = ⟨switch list⟩

Procedure Declarations

⟨formal parameter⟩ : : = ⟨identifier⟩
⟨formal parameter list⟩ : : = ⟨formal parameter⟩|
⟨formal parameter list⟩⟨parameter delimiter⟩⟨formal parameter⟩
⟨formal parameter part⟩ : : = ⟨empty⟩|(⟨formal parameter list⟩)
⟨identifier list⟩ : : = ⟨identifier⟩|⟨identifier list⟩,⟨identifier⟩
⟨value part⟩ : : = ⟨empty⟩|**value**⟨identifier list⟩;
⟨specifier⟩ : : = **string**|⟨type⟩|**array**|⟨type⟩**array**|**label**|**switch**|**procedure**|⟨type⟩**procedure**
⟨specification part⟩ : : = ⟨empty⟩|⟨specifier⟩⟨identifier list⟩;|
⟨specification part⟩⟨specifier⟩⟨identifier list⟩;
⟨procedure heading⟩ : : = ⟨procedure identifier⟩⟨formal parameter part⟩;
⟨value part⟩⟨specification part⟩
⟨procedure body⟩ : : = ⟨statement⟩|⟨code⟩
⟨procedure declaration⟩ : : = **procedure**⟨procedure heading⟩⟨procedure body⟩|
⟨type⟩**procedure**⟨procedure heading⟩⟨procedure body⟩

17.33.7. Classification of Basic Symbols. As stated earlier the Reference language embraces 116 basic symbols.

⟨basic symbol⟩ : : = ⟨letter⟩|⟨digit⟩|⟨logical value⟩|⟨delimiter⟩
⟨delimiter⟩ : : = ⟨operator⟩|⟨separator⟩|⟨bracket⟩|⟨declarator⟩|⟨specificator⟩
⟨operator⟩ : : = ⟨arithmetic operator⟩|⟨relational operator⟩|⟨logical operator⟩|
⟨sequential operator⟩
⟨arithmetic operator⟩ : : = +|−|×|/|÷|↑
⟨logical operator⟩ : : = ≡|⊃|∨|∧|¬
⟨sequential operator⟩ : : = **go to**|**if**|**then**|**else**|**for**|**do**
⟨separator⟩ : : = ,|.|$_{10}$|:|;|: =|␣|**step**|**until**|**while**|**comment**
⟨bracket⟩ : : = (|)|[|]|‘|’|**begin**|**end**
⟨declarator⟩ : : = **own**|**Boolean**|**integer**|**real**|**array**|**switch**|**procedure**
⟨specificator⟩ : : = **string**|**label**|**value**

The separator ␣ denoting a space and the brackets ‘ and ’ are used in forming strings used as parameters of procedures. The separator **comment** is used to introduce explanatory text in an ALGOL 60 program without affecting the meaning of the program; the convention is used that

;**comment**⟨any sequence of basic symbols not containing a semicolon⟩;

is equivalent to a single semicolon.

17.33.8. Example

procedure *Bisection*(*F*)*initial*:(*x1*,*d1*)*precision*:(*d0*)*result bounds*:(*xP*,*xL*);

comment *Finds bounds xP and xL (with difference less than d0) for a zero of the function F(x) at which its derivative is positive. Evaluates F(x) at x1, x1 + d1 and then at equispaced values of x until the zero is passed (indicated by b* = **true**) *when it is located by the method of repeated bisection, xP being the final value and xL the previous value the other side of the zero;* **value** *d0*; **real** *xL, xP*; **real procedure***F*; **begin Boolean***a, b, c*; **real***x, d*; *a* : = *b* : = **false**; *x* : = *x1*; *d* : = *d1*;

A:*c* : = *sign*(*F*(*x*)) = *sign*(*d*);
b : = *c*∧*a*∨*b*;
if *c*∧*a* **then** *xL* : = *x* − *d*;
a : = **true**;
if *b* **then** *d* : = *d*/2;
if *c* **then** *d* : = − *d*;
if*abs*(*d*) ≥ *d0***then begin***x* : = *x* + *d*;**go to***A***end**;
xP : = *x* **end***Bisection*

This process has been used for nonlinear eigenvalue problems in which the largest part of the calculation is the evaluation of the function $F(x)$ and this function involves empirical data so that its derivative is not available. The Boolean variable a is used to distinguish the first passage through the sequence of instructions (when it has the value **false**). The formula for the Boolean variable b is equivalent to $b := (c \wedge a) \vee b$ by the precedence rules for the logical operators. This shows that once b has been assigned the value **true** it will "stick" at that value—indicating that the required zero of $F(x)$ has been "bracketed." The part $c \wedge a$ ensures that this will not happen on the first step (when insufficient information is available for this decision). c = **true** indicates (when a is **true**) that addition of d to x will move it further from the required zero, i.e., that the sign of d is to be changed.

References

1. Adams, Eldridge S., Jr., and Stewart I. Schlesinger: Simple Automatic Coding Systems, *Commun. Assoc. Computing Machinery*, **1**(7):5–9 (July, 1958).
2. Baker, Charles L.: The PACT I Coding System for the IBM Type 701, *J. Assoc. Computing Machinery*, **3**(4):272–278 (October, 1956).
3. Bemer, R. W.: A Checklist of Intelligence for Programming Systems, *Commun. Assoc. Computing Machinery*, **2**(3):8–13 (March, 1959).
4. Boehm, E. M., and T. B. Steel, Jr.: The Share 709 System: Machine Implementation of Symbolic Programming, *J. Assoc. Computing Machinery*, **6**(2):134–140 (April, 1959).
5. Bratman, Harvey, and Ira V. Boldt, Jr.: The Share 709 System: Supervisory Control, *J. Assoc. Computing Machinery*, **6**(2):152–155 (April, 1959)
6. Carr, John: Recursive Subscripting Compilers and List-type Memories, *Commun. Assoc. Computing Machinery*, **2**(2):4–6 (February, 1959).
7. Conway, Melvin E.: Proposal for an UNCOL, *Commun. Assoc. Computing Machinery*, **1**(10):5–8 (October, 1958).
8. Deer, J. I., and R. C. Luke: Semi-automatic Allocation of Data Storage for PACT I, *J. Assoc. Computing Machinery*, **3**(4):299–308 (October, 1956).
9. DiGiri, Vincent J., and Jane E. King: The Share 709 System: Input-output Translation, *J. Assoc. Computing Machinery*, **6**(2):141–144 (April, 1959).
10. Goldstine, Herman H.: Some Remarks on Logical Design and Programming Checks, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., pp. 96–101, Dec. 8–10, 1953.

11. Gordon, Barry: An Optimizing Program for the IBM 650, *J. Assoc. Computing Machinery*, **3**(1):3–5 (January, 1956).
12. Gorman, T. P., R. G. Kelly, and R. B. Reddy: Automatic Coding for the IBM 701, *J. Assoc. Computing Machinery*, **2**(4):253–261 (October, 1955).
13. Gorn, Saul: Standardized Programming Methods and Universal Coding, *J. Assoc. Computing Machinery*, **4**(3):254–273 (July, 1957).
14. Green, Julien: Possible Modifications to the International Algebraic Language, *Commun. Assoc. Computing Machinery*, **2**(2):6–8 (February, 1959).
15. Greenwald, I. D., and H. G. Martin: Conclusions after Using the PACT I Advanced Coding Technique, *J. Assoc. Computing Machinery*, **3**(4):309–313 (October, 1956).
16. Greenwald, I. D., and Maureen Kane: The Share 709 System: Programming and Modification, *J. Assoc. Computing Machinery*, **6**(2):128–133 (April, 1959).
17. Hempstead, Gus, and Jules I. Schwartz; PACT Loop Expansion, *J. Assoc. Computing Machinery*, **3**(4):292–298 (October, 1956).
18. Herbst, Eugene H., N. Metropolis, and Mark B. Wells: Analysis of Problem Codes on the MANIAC, *Mathematical Tables and Other Aids to Computation*, **IX**(49):14–20 (January, 1955).
19. Holt, Anatol W.: General Purpose Programming Systems, *Commun. Assoc. Computing Machinery*, **1**(5):7–9 (May, 1958).
20. Hume, J. N. P., and Beatrice H. Worsley: Transcode: A System of Automatic Coding for FERUT, *J. Assoc. Computing Machinery*, **2**(4):243–252 (October, 1955).
21. Ianov, IU. I.: On the Equivalence and Transformation of Program Schemes, *Commun. Assoc. Computing Machinery*, **1**(10):8–12 (October, 1958). [Originally published *Doklady Akad. Nauk SSSR*, **113**(1):39–42 (1957). Translated by Morris D. Friedman, Lincoln Laboratory.]
22. Lentz, J. J.: A New Approach to Small-computer Programming and Control, *IBM J. Research Development*, **2**(1):72–83 (January, 1958).
23. Lipton, S.: Two Programming Techniques for One-plus-one Address Computers, *J. Assoc. Computing Machinery*, **4**(3):274–278 (July, 1957).
24. Luebbert, William F., and Percy W. Collom, Jr.: Signal Corps Research and Development on Automatic Programming of Digital Computers, *Commun. Assoc. Computing Machinery*, **2**(2):22–27 (February, 1959).
25. Melahn, Wesley S.: A Description of a Cooperative Venture in the Production of an Automatic Coding System, *J. Assoc. Computing Machinery*, **3**(4):266–271 (October, 1956).
26. Miller, Robert C., Jr., and Bruce G. Oldfield: Producing Computer Instructions for the PACT I Compiler, *J. Assoc. Computing Machinery*, **3**(4):288–291 (October, 1956).
27. Mock, Owen, and Charles J. Swift: The Share 709 System: Programmed Input-output Buffering, *J. Assoc. Computing Machinery*, **6**(2):145–151 (April, 1959).
28. Mock, Owen: Logical Organization of the PACT I Compiler, *J. Assoc. Computing Machinery*, **3**(4):279–287 (October, 1956).
29. Naur, Peter (ed.): Report on the Algorithmic Language ALGOL 60, *Commun. Assoc. Computing Machinery*, **3**(5):299–314 (May, 1960).
30. Perlis, A. J., and K. Samelson: Preliminary Report—International Algebraic Language, *Commun. Assoc. Computing Machinery*, **1**(12):8–22 (December, 1958).
31. Rowan, T. C.: Psychological Tests and Selection of Computer Programmers, *J. Assoc. Computing Machinery*, **4**(3):348–353 (July, 1957).
32. Scott, A. E.: Automatic Preparation of Flow Chart Listings, *J. Assoc. Computing Machinery*, **5**(1):57–66 (January, 1958).
33. Shell, Donald L.: The Share 709 System: A Cooperative Effort, *J. Assoc. Computing Machinery*, **6**(2):123–127 (April, 1959).
34. Sheridan, Peter B.: The Arithmetic Translator-compiler of the IBM Fortran Automatic Coding System, *Commun. Assoc. Computing Machinery*, **2**(2):9–21 (February, 1959).
35. Steel, T. B., Jr.: PACT IA, *J. Assoc. Computing Machinery*, **4**(1):8–11 (January, 1957).
36. Tillitt, Harley: Computer Programming for Young Students, *J. Assoc. Computing Machinery*, **5**(4):309–318 (October, 1958).
37. Voorhees, Edward A.: Algebraic Formulation of Flow Diagrams, *Commun. Assoc. Computing Machinery*, **1**(6):4–8 (June, 1958).
38. Wagner, Harvey M.: A Lemma for Automatic Optimum Programming on the IBM 650, *Mathematical Tables and Other Aids to Computation*, **XI**(58):101–104 (April, 1957).
39. Wegstein, J. H.: From Formulas to Computer Oriented Language, *Commun. Assoc. Computing Machinery*, **2**(3):6–8 (March, 1959).
40. Yates, F., and S. Lipton: An Automatic Programming Routine for the Elliott 401, *J Assoc. Computing Machinery*, **4**(2):151–156 (April, 1957).

Section 18

INPUT AND OUTPUT

By

MORTON M. ASTRAHAN, *Systems Engineer, International Business Machines Corporation, Advanced Systems Development Division Laboratory, San Jose, Calif.*

LOWELL S. MICHELS, *Principal Engineer, Bendix Pacific Division, The Bendix Corporation, North Hollywood, Calif.*

WILLIAM A. FARRAND, *Staff of Chief Engineer, Autonetics Division, North American Aviation, Inc., Downey, Calif.*

CONTENTS

18.1. Punched Paper Tape 18–1
18.1.1. Physical Description 18–2
18.1.2. Reading 18–4
18.1.3. Punching 18–5
18.2. Line Printers 18–6
18.3. Punched-card Input and Output 18–8
18.3.1. Reading 18–8
18.3.2. Punching 18–12
18.4. Magnetic-tape Input and Output Devices 18–13
18.4.1. Analog Tape Devices 18–13
18.4.2. Digital Magnetic-tape Equipment 18–14
18.4.3. Converters to Magnetic Tape 18–15
18.4.4. Converters from Magnetic Tape 18–15
18.4.5. Special Magnetic Input and Output Devices 18–18
18.5. Symbol Displays 18–20
18.5.1. Selectively Positioned Symbols 18–21
18.5.2. Selectively Illuminated Symbols 18–22
18.5.3. Cathode-ray Methods 18–25
18.6. Analog-to-digital Converters 18–29
18.6.1. Voltage-to-number Converters 18–30
18.6.2. Position-to-number Converters 18–35
18.6.3. Time and Frequency 18–41
18.7. Digital-to-analog Converters 18–42
18.7.1. Number-to-voltage Converters 18–43
18.7.2. Number-to-position Converters 18–46
18.7.3. Time and Frequency 18–49
18.8. Plotters 18–50
18.8.1. Writing Means 18–50
18.8.2. Displacement Control 18–53
18.8.3. Cathode-ray Techniques 18–57

18.1. PUNCHED PAPER TAPE*

The perforated-tape storage medium has been in use for many years. As far back as 1833 in Europe, Gauss and Weber invented a five-unit code to replace the old

* By Morton M. Astrahan.

semaphore system of communication. Later contributions in the nineteenth century were made by Hughes and Baudot. The present teletype system, using five-channel paper tape based on the five-unit Baudot code, dates back to experiments begun in 1902.

The virtues of the perforated tape as an economical storage medium have been noted by many. As a result, the code system has expanded to use more than five code holes. Generating equipment now produces five-, six-, seven-, or eight-channel perforated tape. Five-channel tape is most widely used because of its adaptability to transmission on standard equipment; it is the common-language tape and continues to grow in popularity. Use of perforated tape for six-, seven-, or eight-channel information is much less frequent.

The common-language five-channel tape is generated as chad or chadless tape. Chips are completely removed from chad tapes forming a clean hole, while chadless tape contains a hinged chip so that interpretation printing can occur on the same tape (see Figs. 18.1*a* and 18.1*b*). In general, a keying operation will directly produce

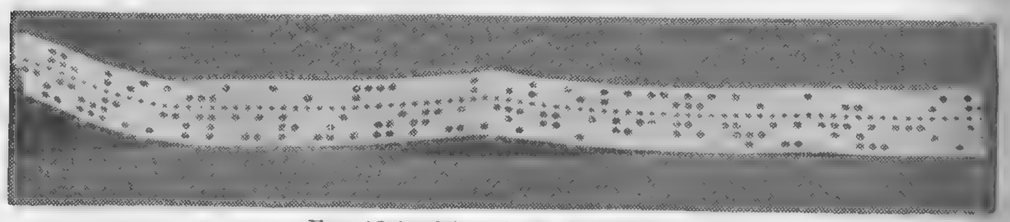


FIG. 18.1*a*. Five-channel chad tape.

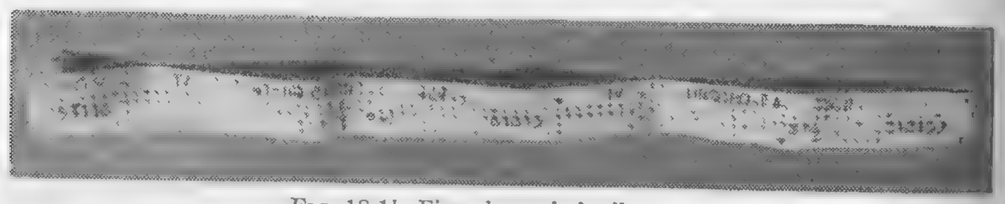


FIG. 18.1*b*. Five-channel chadless tape.

a chad tape. Tape reperforated following wire transmission will be chadless. Interpretation of chadless tape, which is done at the same time as punching, serves as visual instruction in a communications switching center.

18.1.1. Physical Description. The material used in the tape is usually a 0.004-in.-thick wood-sulfide paper of good quality, which has been impregnated with about 12 per cent mineral oil. For increased mechanical strength, 100 per cent rag bond, 100 per cent rag fiber, or plastics may be used, but the life of the punching mechanism may be affected.

The paper can be purchased in fanfolded form or in a 1,000-ft reel capacity. Most generating equipment employs a blank reel of paper tape.

Paper-tape widths vary with manufacturer but are approximately as follows:

5 channel	0.686 in.
6, 7 channel	0.875 in.
8 channel	1.000 in.

The tapes will contain from five to eight recording positions used in various combinations to represent letters, numbers, special symbols, and control signals. The recording is made by punching holes in the appropriate positions in one column across the width of the tape. These holes in present equipment are circular and are referred to as "code holes." Another, usually smaller hole, is punched in each column to provide a means of accurately moving the tape from column to column along its length. Like the code holes, the "feed hole" or "sprocket hole" is circular. Figure 18.2 shows the Electronic Industries Association standard dimensions as of October, 1959, for 1-in. tape for six, seven, or eight channels. Figure 18.3*a* shows the teletype code for five-channel tape. A good survey of punched media is given in Ref. 33.

Applications. Communication machines are major users of punched tape. Transmission of data to and from outlying plants or warehouses produces tapes that can be used in data-processing systems. Typewriter tape punches, such as the Flexowriter, are available, wherein the tape is produced directly or as a by-product of typing the original document. It can then be used for input to a data-processing

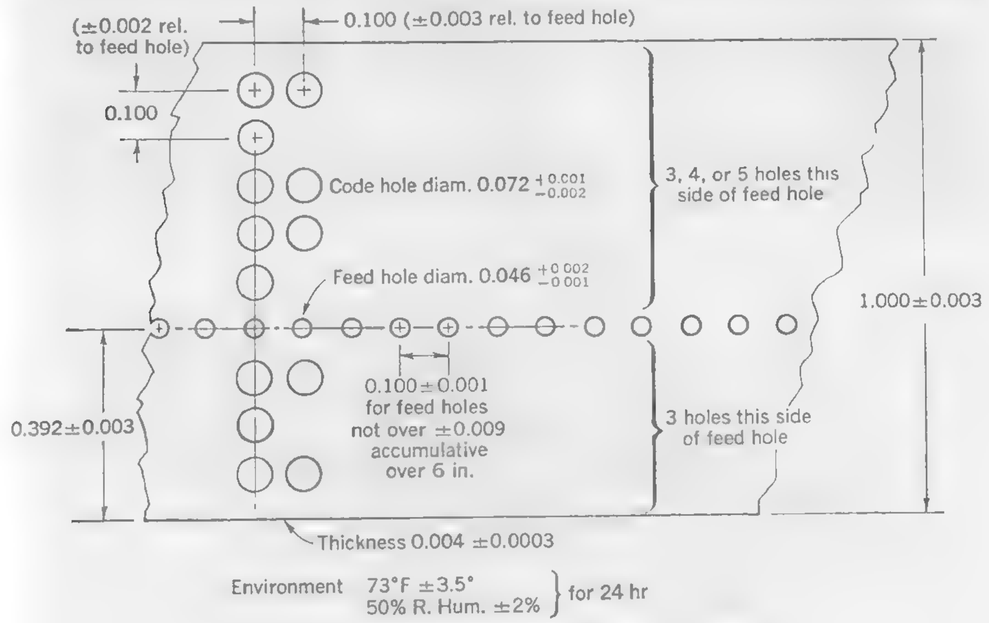

(±0.002 rel. to feed hole)
0.100 (±0.003 rel. to feed hole)
0.100
Code hole diam. 0.072 +0.001 −0.002
Feed hole diam. 0.046 +0.002 −0.001
3, 4, or 5 holes this side of feed hole
1.000±0.003
0.100±0.001 for feed holes not over ±0.009 accumulative over 6 in.
0.392±0.003
3 holes this side of feed hole
Thickness 0.004 ±0.0003
Environment 73°F ±3.5° 50% R. Hum. ±2% for 24 hr

FIG. 18.2. EIA Standard RS-227 for 1-in. tape.

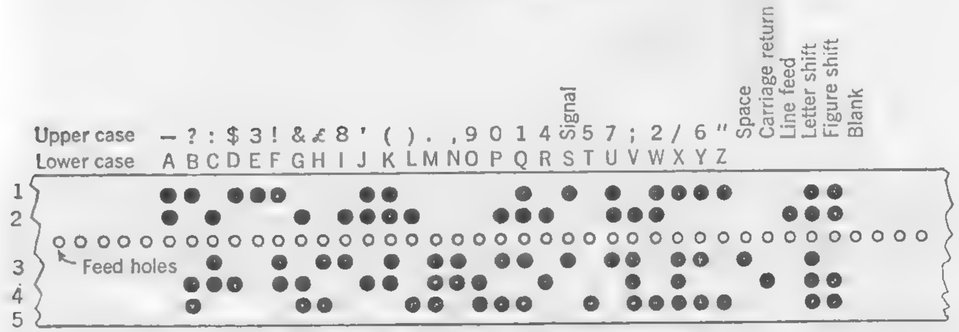

Upper case — ? : $ 3 ! & ¢ 8 ' ( ) . , 9 0 1 4 Signal 5 7 ; 2 / 6 " Space Carriage return Line feed Letter shift Figure shift Blank
Lower case A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1
2
3
4
5
Feed holes

FIG. 18.3*a*. Five-channel teletype code.

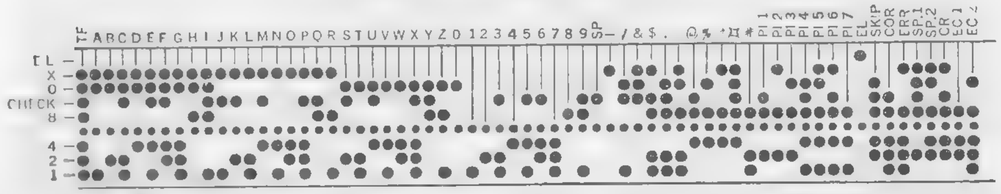

TF A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 SP − / & $. @ % * ¤ # PI 1 PI 2 PI 3 PI 4 PI 5 PI 6 PI 7 EI EK P COR ER1 SP1 SP2 COR EC1 EC2
TL
X
0
CHECK
8
4
2
1

FIG. 18.3*b*. Eight-channel code for tape-to-card conversion. (*Courtesy of IBM Corporation.*)

system, for conversion to punched cards for subsequent use in punched-card accounting systems, for transmission, or for semiautomatic typing of succeeding documents. Accounting, bookkeeping, adding, and calculating machines are being provided with punched-tape output to eliminate subsequent key-stroke operations for the preparation of records to be handled in a data-processing system. The detection and correction of operator errors are one of the major problems in such systems.

Many computer systems accept punched tape directly as an input. In others, the tape is first converted to punched cards. Figure 18.3*b* shows the eight-channel code used with the IBM 046 tape-to-card punch. The check channel provides an odd parity count for each character. The control characters, such as End Card 1 (EC1) or Program Level 1 (PL1), are instructions to the tape-to-card punch.

18.1.2. Reading. Mechanical pin-sensing readers cover the range from 6 to 60 characters per second. The Teletype 60-character-per-second reader is shown schematically in Fig. 18.4. Figure 18.4 shows one of the five sensing pins, along with its associated read-out contacts. The eccentric drive and main bail are common to all sensing positions.

A vertical reciprocating type of motion is transferred to the main bail by means of the bail follower and eccentric drive. On the upstroke, the tape is sensed; on the downstroke, the tape is advanced to bring the next character into the sensing position.

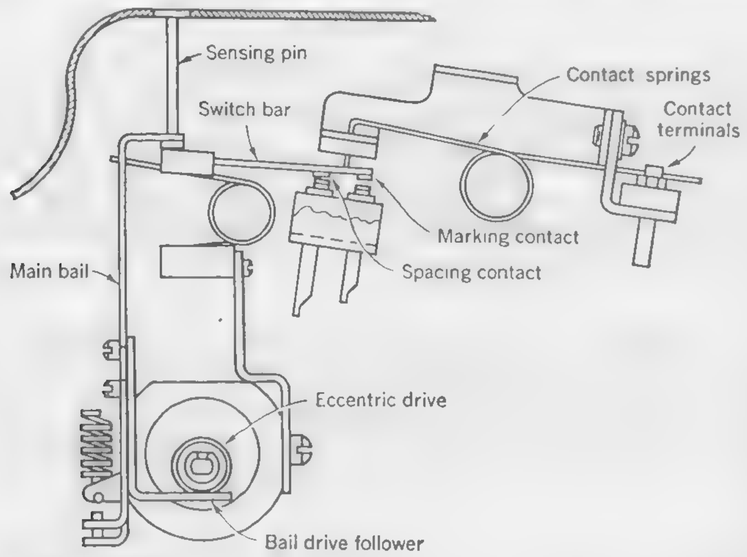


FIG. 18.4. Tape-sensing mechanism. (*Courtesy of Teletype Corporation.*)

The sensing pin extends through a hole in the main bail and rests on the switch bar's insulated extension. At the other end of the switch bar is a contact assembly consisting of marking and spacing contacts. The sensing pin is shown in the down position, having been pulled there by the main bail. In this position the switch bar makes contact with the spacing contact.

As the eccentric drive starts the upstroke, the main bail is allowed to move up under the tension of the main bail return spring. When the bail moves, it allows the sensing pins to move up under the force of the sensing pin springs. If there is a hole in the tape, the sensing pin continues its upward motion, passing into the hole. The switch bar, which follows the sensing pin, rocks from the spacing contact to the marking contact as the sensing pin enters the hole in the tape. This completes a circuit from the contact spring, through the switch bar and out the marking contact, putting out a marking pulse.

If there is no hole in the tape, the upward motion of the sensing pin is halted by the tape and the switch bar remains on the spacing contact.

On the downstroke, the main bail returns all sensing pins, switch bars, and contacts to the spacing position and advances the tape. The tape-feed mechanism is basically a pawl-and-ratchet arrangement.

An interesting feature of the feeding operation is that the sensing pins are still projecting through the holes at the beginning of the tape advance. The sensing pins yield to the forward motion of the tape until they are withdrawn from the holes, at which time they snap back to their normal position, ready to sense the next character.

Photoelectric sensing provides a faster means of reading tape. Commercially available photoelectric readers for fully punched tape read from 50 to 1,500 characters per second. The tape is guided past a mask which contains apertures of the size and spacing for each possible hole in a row. Illumination through the holes in the punched tape, from a constant light source, is gathered optically and focused on photocells. Electrical signals are generated in the photocells, when a punched hole passes over the mask aperture. In addition to making possible greater reading speeds, photoelectric sensing eliminates one source of tape wear, since there are no pins or fingers pressing against the tape. The signals generated in the photocells, when amplified to usable levels and decoded, are in a form convenient for computer input.

18.1.3. Punching. Electromagnets actuating mechanical punches have been employed in tape-punching equipment. The inertia inherent in the mechanical mechanism restricts punching speeds to several characters per second.

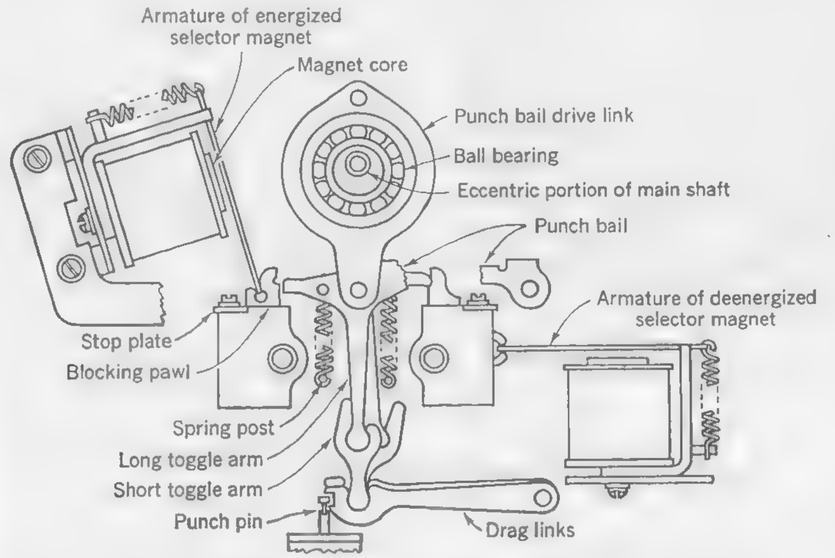


FIG. 18.5. Punching mechanism. (*Courtesy of Teletype Corporation.*)

Figure 18.5 shows a schematic of a Teletype 60-character-per-second motorized punch. A similar mechanism now operates at 110 characters per second. The figure shows two punching mechanisms, one behind the other, with their respective selector magnets.

The eccentric main shaft generates a continuous 60-cycle-per-second reciprocating vertical motion which is transmitted to the punch bail. This motion is then transmitted from the bail to all toggle arms, drag links, and finally to the punch pins. The total peak-to-peak vertical displacement is 0.090 in.

The eccentric main shaft is shown in the lowermost position. Note that the long toggle arm of the outer mechanism has cleared its blocking pawl as a result of the energizing of the selector magnet spring, and that the long and short toggle arms behave as one stiff member, fully transferring the displacement of the eccentric to the punch pin. This complete displacement is sufficient to drive the punch pin through the paper tape.

The selector magnet for the other mechanism is deenergized, causing its blocking pawl to block the long toggle arm. This causes the long toggle arm to rotate slightly counterclockwise, throwing the joint between the long and short toggle links to the right as shown. Since part of the drive displacement is taken up in collapsing the toggle joint, the punch-pin is not driven a sufficient distance to puncture the paper tape.

Identical linkages are used to operate the tape-feed mechanism. Tape feeding is also selective, so that tape is fed only when a character has been perforated.

Punches are now available with speeds up to 300 characters per second.

18.2. LINE PRINTERS*

Among various computer output devices, the line printer is perhaps the most important. This is more true today, when the emphasis is on data processing, than it was in the early 1950s when the emphasis was chiefly on scientific computation.

The line printer, as its name implies, is capable of printing a complete line of information for each machine cycle. Characters may be either numeric or alphabetic. At the present time, the available equipment provides a variety of line lengths, ranging from 48 to as many as 130 characters per line.

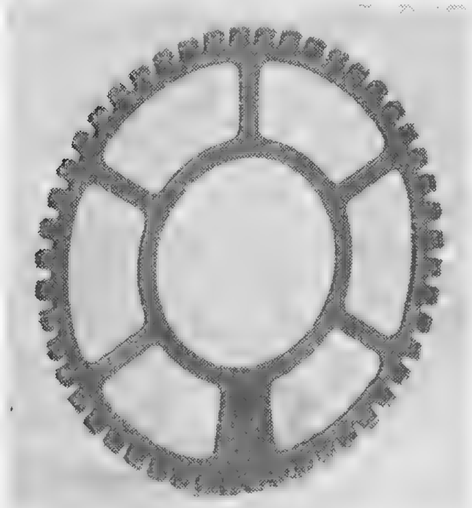


FIG. 18.6*a*. Print wheel from IBM 407 accounting machine printer. (*Courtesy of IBM Corporation.*)

The first line printers evolved from the punched-card tabulator. These printers are primarily mechanical and produce 80 to 150 lines per minute. The wheel printer is an example of this type of line printer.[1]

The mechanical wheel printer uses a type wheel for each character position and operates through mechanical setup of the wheel for each character to be printed. A typical type wheel for the IBM 407 accounting machine is illustrated in Figs. 18.6*a* and 18.6*b*. This mechanism is used in many IBM 700 series systems. Each print wheel starts to rotate at a time determined by the character to be printed. The motion of the wheel against the ribbon and paper cancels the rotation. The result is that the print wheel is, in effect, momentarily stopped at the time the wheel moves against the paper. Alphabetic, numeric, and symbolic characters are arranged around the periphery of the wheel.

The next developments for the 500- to 1,000-line-per-minute speed range were the so-called "on-the-fly" printers and wire printers. "On-the-fly" printers are primarily electronic, but the actual printing is done when a hammer strikes a rotating print wheel. An electronic store for all the characters of the line to be printed is needed.

FIG. 18.6*b*. Close-up of print wheel showing type. (*Courtesy of IBM Corporation.*)

The print wheels are arranged on a horizontal shaft, or the characters are on a cylindrical surface equivalent to a stack of print wheels. The angular position of the cylinder is monitored by a timing mechanism, usually optical. A hammer for each character position is fired at the appropriate point in the cylinder revolution. Many printers are based on variations of this method.[16]

The wire printer makes use of a matrix of 35 wires to set up the character to be printed. Figure 18.7 shows the matrix arrangement as well as the layout of wires from the code rod to the print head. Binary-coded decimal information is used to cause rotational and vertical movement of the code rod. There are seven rotational positions of the code rod and eight vertical positions. The combination results in

* By Morton M. Astrahan.

56 possible character configurations. The IBM 720A printer reaches 500 lines per minute using 30 heads and 4 printing positions per head for 120-character lines.

Burroughs makes an electrostatic printer in which a charge pattern is deposited by styli moving across high-resistance paper.[9] The printing is made visible by a toner which is attracted to the charged areas and thermally fixed.

The IBM 370 printer is a 50-character-per-second serial printer using a character stick which moves incrementally across the page. The stick is rotated and translated vertically by metal bands for character selection.

More recent printer developments include the chain printer and the Charactron printer.

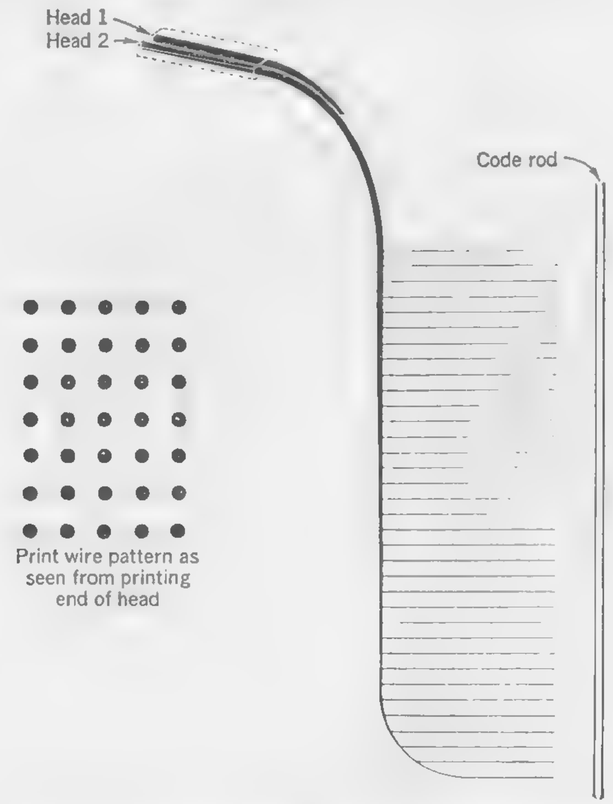


FIG. 18.7. Wire-printer principle. (*Courtesy of IBM Corporation.*)

One problem with the cylindrical "on-the-fly" printers is the difficulty of getting the printed characters to lie exactly on a horizontal line, partly because of variations in hammer characteristics. On the theory that such vertical misalignment is more noticeable than horizontal misalignment, the chain printer moves the characters horizontally on a flexible band. A hammer is fired in each character position when the appropriate character goes by. The IBM 1403 prints 600 132-character lines per minute.

The Charactron printer uses a special cathode-ray tube to generate a line of characters on a phosphor screen. The writing electron beam is shaped by passage through character-shaped openings in a mask. The output, optically coupled to the screen, can be microfilm or, with xerographic techniques, full-sized printing on paper. Speed depends on format and is claimed to be over 4,000 100-character lines per minute with the Stromberg-Carlson 5000 printer, and up to 15,000 characters per second on microfilm with the SC 4000.

The use of any of the equipment discussed above in a computing system involves problems of synchronization, translation, and buffering. Because of the mechanical

nature of the printers, these machines are asynchronous with the internal computer timing cycle.

Although it is not too difficult to tie directly into the computer, thus eliminating the need for external buffering, it is not generally desirable to delay computing operations for the relatively slow transfer of information to the output printer. In some cases also, the information from the computer is available only serially by character, while most line printers require a parallel input. A buffer may serve not only for temporary storage of information, but is also capable of effecting a serial-to-parallel conversion. The magnetic core is ideally suited for use as a memory element in such a buffer, since a matrix can readily be wired to provide the necessary conversion. These considerations also apply to the use of a buffer in a separate magnetic-tape-to-printer operation.

18.3. PUNCHED-CARD INPUT AND OUTPUT*

Punched-card input and output equipment for electronic digital computers has evolved from the standard card-handling mechanisms which have been employed extensively in electromechanical accounting equipment. The standard card allows high-speed input and output in a form which can also be prepared and further processed by the large variety of existing punched-card accounting machines. Existing files of cards can also be used.

The cards are key-punched from source data or are punched automatically as by-products of bookkeeping-machine operations, of factory data-gathering systems, or of other source-recording terminals, such as character-reading systems. They are available in many special combinations for use as source documents. Systems are available for transmitting them over telephone circuits.

The most widely used card is the 80-column card shown in Fig. 18.8*a*. There are 12 punching positions in each column and information is normally punched one character to a column, as shown in Fig. 18.8*a*. The cards can also be punched with binary information, as shown in Fig. 18.8*b*. This greatly increases the density of information storage. Reference 33 shows other types of punched cards.

For some uses it is satisfactory if the card-reading and -punching mechanisms are connected directly to the computer. For faster information transfer, separate card-to-tape machines are used. These allow the information from many cards to be stored on magnetic tape and later be entered into the computer very rapidly. Cards punched with decimal or alphabetic characters can be prepared on commercially available card-punching machines. Both programs and data can be entered in this way. With a binary computer, a translation program must be used to convert the information from decimal to binary.

Binary punched cards are often used for programs on binary machines and are prepared by the computer. Automatic-programming techniques take statements on cards in alphanumeric form and provide the detailed program on binary cards or on magnetic tape.

18.3.1. Reading. Card-handling mechanisms are designed to pick one card from the bottom of a stack, move it past a sensing or punching station, and deposit it on another stack. A reading mechanism is shown in Fig. 18.9. The cards are held in a hopper under a weight. A pair of picker knives with shoulders slightly lower than the thickness of a card push the bottom card through a throat which is slightly higher than the thickness of one card. The card is caught by feed rollers and moved past the reading stations. The picker knife motion is controlled by a clutch, allowing selective reading. Holes are sensed by stranded metal brushes riding on a metal roll. The card passes between the brushes and the roll and allows electrical contact where holes are punched. The card is moved edgewise (the direction of motion is perpendicular to the long dimension) and there is a brush for each of the 80 columns. The card motion is accurately timed in relation to cam-operated contacts, so that signals are available to mark the points at which each of the 12 rows of punching positions is under the brushes.

* By Morton M. Astrahan.

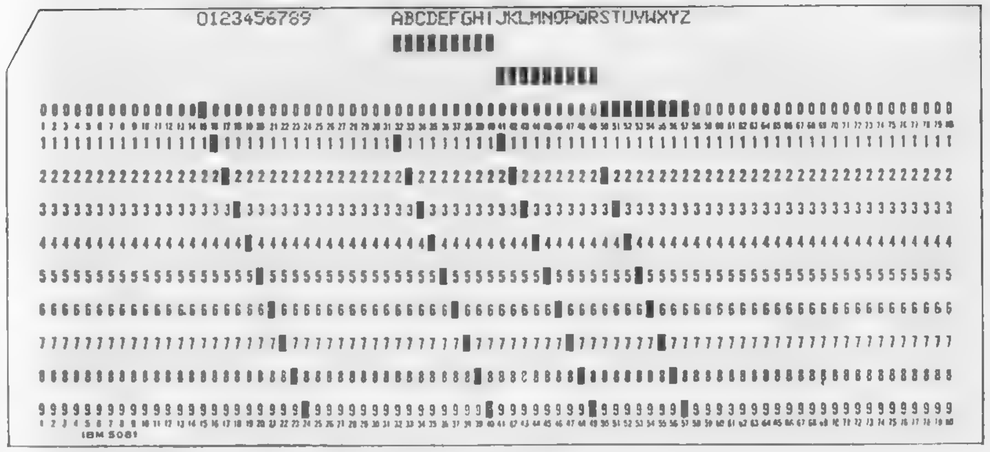


FIG. 18.8*a*. Alphanumeric card with IBM code.

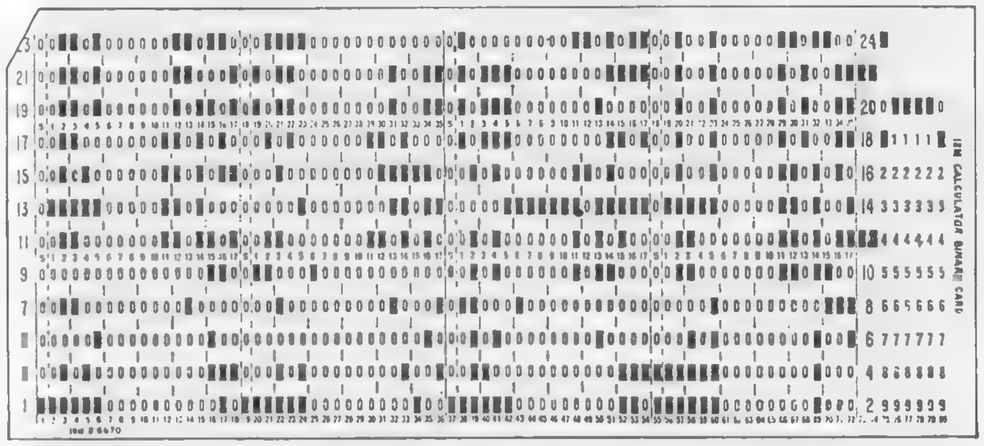


FIG. 18.8*b*. Binary card.

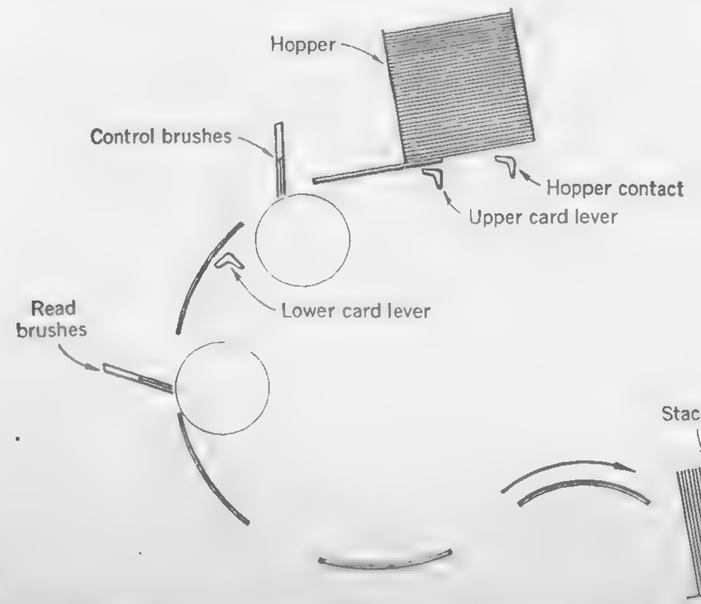


FIG. 18.9. Type 711 card-reader schematic. (*Courtesy of IBM Corporation.*)

Figure 18.10 shows the card flow through the IBM Type 1402 Card Read-Punch which reads cards at a maximum speed of 800 cards per minute. At this speed the limited capacity of a standard hopper would result in excessive operator attention. Therefore, a file feed is used. This consists of a portable tray holding up to 3,000 cards and a mechanism for automatically feeding cards from the tray into the hopper. The Type 1402 has five card-stacker pockets. Cards from the read feed can be directed into any one of the three pockets nearest the read feed. Figure 18.11 shows a schematic of the radial card stackers. The selection mechanism stops the card's

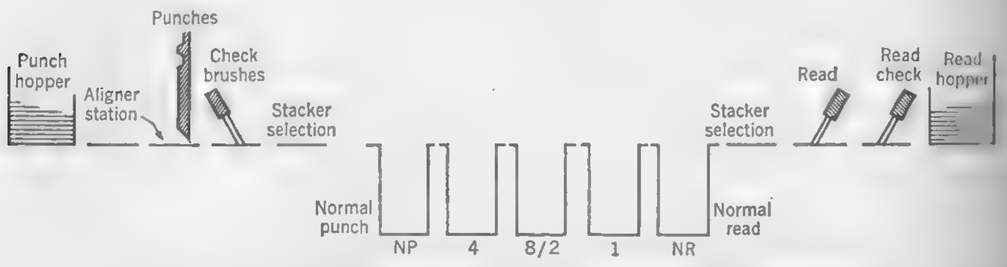


FIG. 18.10. 1402 card-feed schematic. (*Courtesy of IBM Corporation.*)

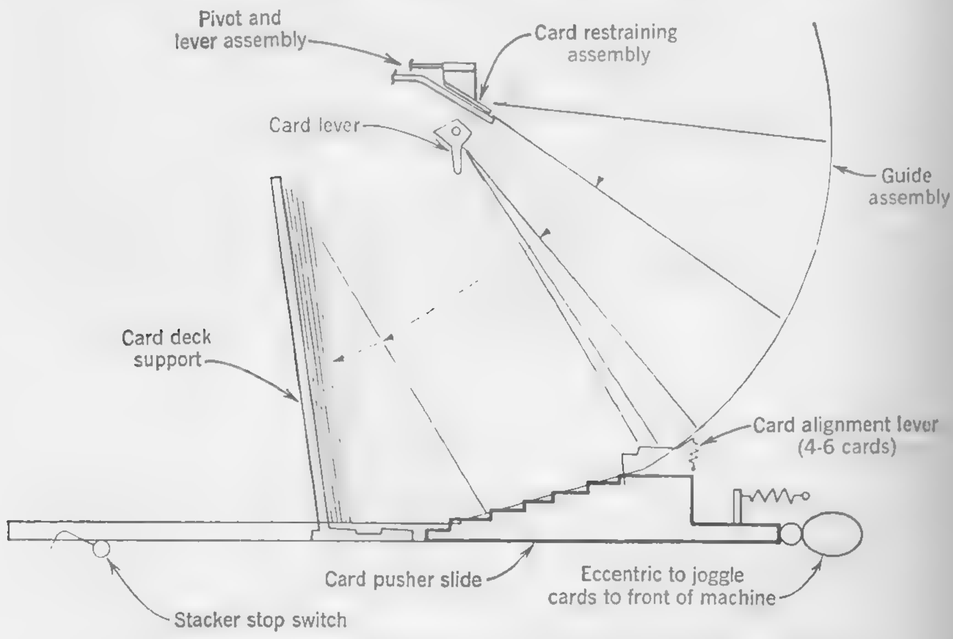


FIG. 18.11. 1402 stacker schematic. (*Courtesy of IBM Corporation.*)

forward motion and drops it into a stacker in a horizontal position. The rear end of the card falls, guided by the guide assembly. The radius of the guide assembly is such that as the rear of the card approaches the bottom, the front falls from the card restraining lever that has been supporting it. The card rests temporarily on the card alignment lever until four to six cards have accumulated. Their weight overcomes the lever spring tension and the group of cards is lowered onto the card pusher slide. The pusher slide oscillations work the bottom of the cards forward until they are standing against the card deck support.

Long lines are usually needed from the reading brushes to the sensing circuits, and these lines usually are physically near relay circuitry. Thus noise pickup is high and the sensing circuits should be low-impedance and insensitive to short (relative to

msec) pulses. Pencil marks on cards can cause errors in circuits with impedances above several thousand ohms, by giving rise to current paths between brushes. The sensing circuit must reliably respond to ragged pulses of less than normal duration resulting from the brush bounce and dirt to be expected after long periods of operation.

Figure 18.12 shows the type of circuit used for sensing in early computer systems. The brushes were connected through a plugboard to key triggers. These were flip-flop circuits which were desensitized by means of a condenser between the grids of the tubes so that a pulse of at least 1-msec duration was required to set them. The pulse from the brushes was cam-controlled so that current interruption took place at a cam contact rather than at the card. The full pulse time (about 9 msecs, at 150 cards per minute) was allowed to elapse before the state of the key triggers was sampled and the information sent to the high-speed memory.

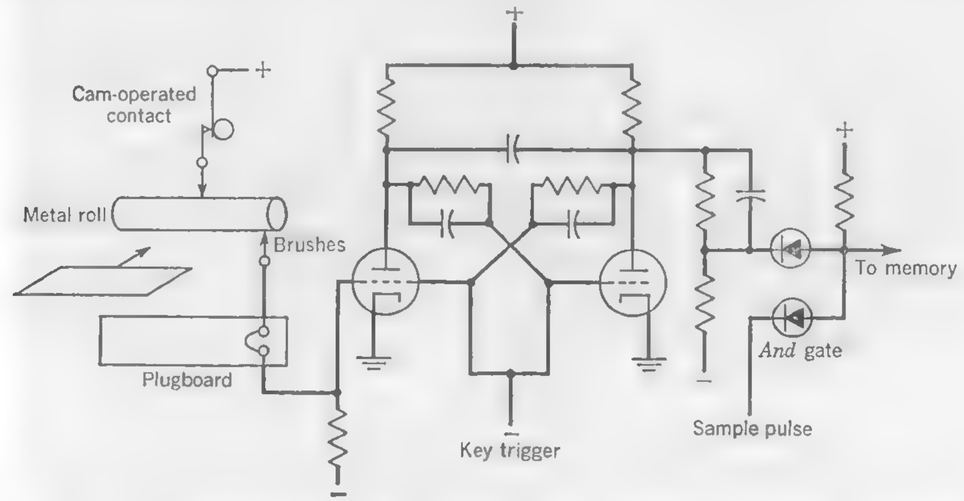


FIG. 18.12. Card-sensing circuits, IBM 701–704 Data Processing Systems. (*Courtesy of IBM Corporation.*)

Thirty-six brush outputs were sampled at once and entered into memory as a 36-bit binary word. Two such words were read from each row of the card, making a total of 24 words covering 72 of the 80 card columns. Binary information was utilized directly. Decimal or alphabetic characters were translated by means of a program which operated in the interval between the sampling of successive rows.

Card-feeding mechanisms which move the card endwise (motion perpendicular to the short dimension) are also in use for computer input. An example is the first Univac card-to-tape converter.[1] In this system the cards were scanned a column at a time by a block of 12 photocells. The photocell outputs conditioned thyratron control grids, and sampling was done by a pulse on the screen grids. The 12 thyratron outputs were encoded into eight lines which caused a character to be recorded on the magnetic tape. The NCR Type 380 feeds cards endwise at a speed of 1,500 cards per minute.

Another Univac card-to-tape machine uses edgewise feed and brush sensing. Here a contact surface is impulsed for each card row and touches individual brushes through the card holes. Each brush feeds one column of an 80-column magnetic core array. The array has 12 cores per column, so that there is a core for each of the 960 possible holes. The proper row in the array for each card row is selected by switching of bias currents. Thus each punched hole causes the corresponding core to be switched by coincidence of the brush current and the proper bias current. The array columns are then read by current pulses applied to each column in sequence through a plugboard, which allows a format control. Bias currents may also be switched during

readout to prevent "control" rows in selected columns from reading out until a later readout signal is fed to the same column, thus permitting column splits in which control information is entered in the same column as an alphanumeric character. The characters are read by sense wires in each row, translated from card code to machine code, and stored on magnetic tape. The tape record is then reread and checked against a second reading of the card.

Cores are widely used for input-output buffering. They are small, economical, reliable, and consume no power when not active. They provide the signal integration needed for noise protection. They also provide the parallel to serial conversion that is usually required with edgewise card feeds.

The highest speed readers use optical sensing. For example, the Uptime 2000, which handles up to 3,000 cards per minute, uses optical sensing both for reading the holes and for timing the card position.

18.3.2. Punching. A widely used mechanism for punched-card output is the 100-card-per-minute punch used with the IBM 700 series systems. One card row is punched at a time. A separate solenoid for each of the 80 columns can move an interposer into the path of a bail, forcing a punch through the card into a die. The card is at rest during each of the 12 punching cycles, the intermittent motion being supplied by a geneva mechanism.

Card punching requires even more precise control of the card position than does reading. For this reason, the punch feed of the IBM Type 1402 Card Read-Punch, which operates at 250 cards per minute, has an alignment station. At this station the card is momentarily released from the drive rolls and is pushed into precise registration with the punching mechanism. The card is then advanced intermittently through the punching station under control of the geneva mechanism shown schematically in Fig. 18.13. When the geneva pins ride into the deep cuts in the geneva disks,

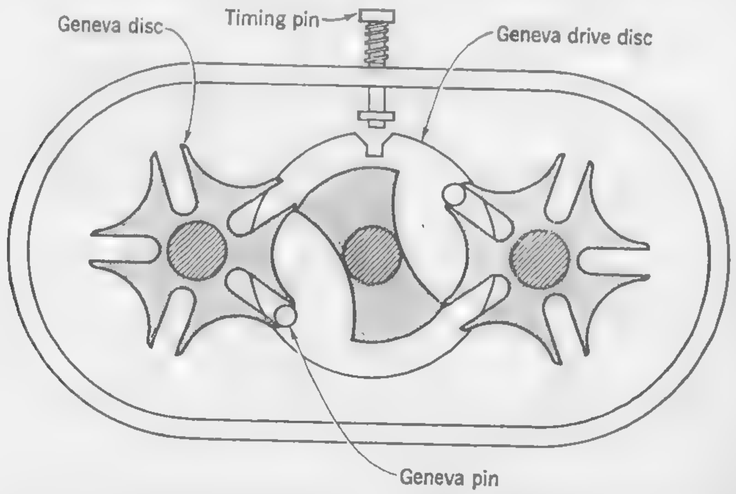


FIG. 18.13. Geneva mechanism. (*Courtesy of IBM Corporation.*)

the feed rolls are driven by gears attached to the geneva disks. As the geneva pins leave the deep cuts, the cam surfaces of the geneva drive disk enter the shallow cuts on the geneva disks, holding the disks and their associated feed rolls stationary.

The punching station operates on one card row at a time by means of 80 units like the one shown schematically in Fig. 18.14. The bails are common to all units and punching is caused by the extension of an interposer between the oscillating punch bail and a punch. However, the interposer motion is not caused directly by the magnet armature motion. Instead, the armatures are closed by latches pulled down by the interposer setup bail. A small holding current is applied to those magnets which are to cause punching. The armatures of these magnets hold their latches down when the setup bail moves up. This causes the associated interposer links to

pivot, extending the interposers. Since the magnet holding current is much lower than that which would be needed to close an armature against a load, the unit is called a "no-work-magnet" unit.

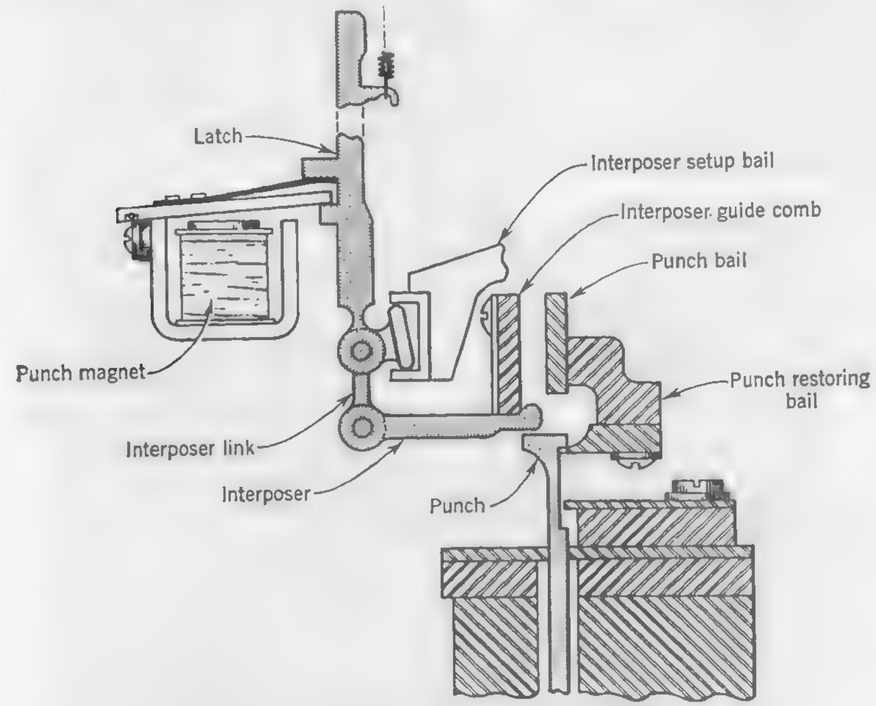


FIG. 18.14. Punch unit operation. (*Courtesy of IBM Corporation.*)

18.4. MAGNETIC-TAPE INPUT AND OUTPUT DEVICES*

The use of magnetic tape for the recording of audio signals is well known. Tape recorders are extensively used in the radio industry, in homes, and as analog-recording devices in real-time processes. Digital computers have used magnetic tape for auxiliary memory devices and for input/output media almost from the time of the development of the first automatic calculator in the late 1940s.

As is indicated above, magnetic-tape units have been extensively used in both the analog- and digital-computer fields. The nature and requirements of these applications are sufficiently different to warrant separate descriptions.

18.4.1. Analog Tape Devices. Of the two applications (analog and digital) the requirements of the analog systems are more like the requirements encountered in high-fidelity audio tape recorders. Indeed, although the analog tape units have many special requirements, they have usually incorporated the techniques used in audio work (see Fig. 18.15).

Data Recording. The principal use of magnetic tape in analog work is the recording of data produced by a real-time system for later input to some type of analog-computer equipment or analog-to-digital converter. The real-time system quite frequently involves a telemetering link between an airplane or missile and a ground station. The frequencies to be recorded in such an application are usually in the audio range, less than 10 kc. However, unlike broadcast and analogous work, these signals are not usually recorded directly on tape. Instead, they are presented to the tape unit as an FM carrier in the ultrasonic range (between 50 and 100 kc) in order to obtain the accuracies desired.

Playback of the data so recorded presents a problem because the accuracy of the data read from the tape is dependent upon the frequency of the carrier being read

* By L. S. Michels.

from the tape, which in turn is a function of the speed of tape transport in both the recorder and playback unit. Since accuracy is of primary interest, elaborate equipment is employed to control the speed of the tape. Although tape-drive mechanisms of high quality are employed to reduce wow, an additional timing track must be used for tape-speed control. In a typical system, a 60-cycle signal obtained from a tuning fork is used to amplitude-modulate an 18-kc carrier which is recorded on the timing track at the same time as the analog signal. On playback, the modulated signal is compared with the tuning-fork frequency by means of a differential motor which drives the tape-movement capstan. Accuracies of the order of 0.02 per cent variation in carrier frequency can be obtained in this manner.

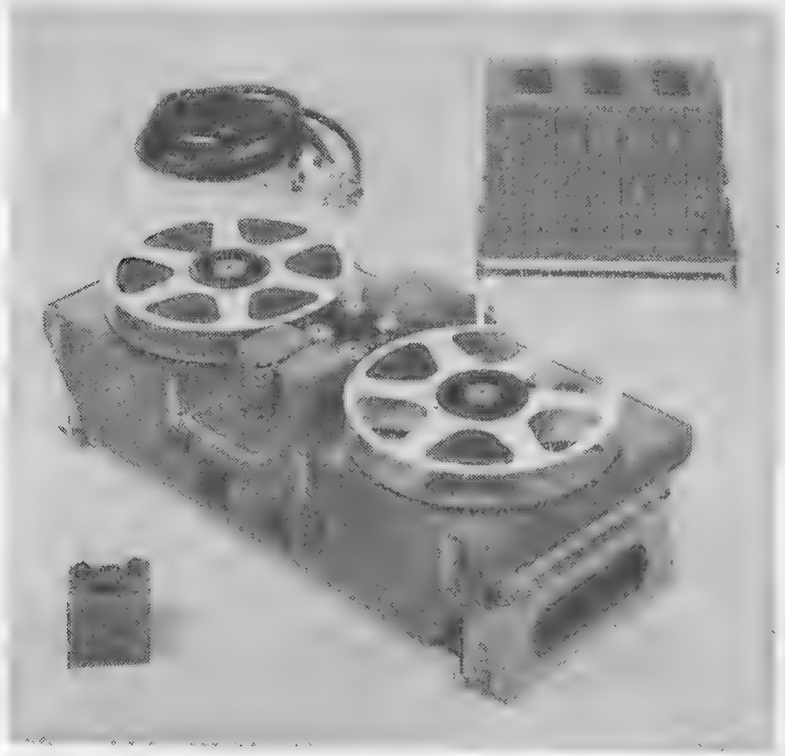


FIG. 18.15. An analog magnetic-tape unit.

Other Applications. Magnetic tape is used, but not extensively, for function input on analog computers. A variation of this procedure is the use of tape for function storage or function delay, in which an output of an analog computer is recorded on tape and later read to become a function input to the computer.

Characteristics. Analog tape units usually employ a number of channels all recorded simultaneously but with different data on each track. Tape speeds vary from approximately 7 to 100 in. per sec, with 30 and 60 in. being common speeds. Many of the recorders are multispeed devices with a very fast rewind speed. Tapes are rarely read in reverse direction.

18.4.2. Digital Magnetic-tape Equipment. The operational requirements of tape equipment for use in digital applications differ markedly from those of analog equipment. The applications are quite similar to that of the analog tape equipment, however, there being two basic uses of digital tape equipment:

1. As auxiliary memory devices
2. As input and output media

Memory Systems. In the section on memory systems various types of memories have been mentioned. In all these there is a practical limit to the amount of information which can be stored, this limit being determined by economic considerations.

In magnetic-drum-type memories, for example, the limit is in the order of a few million characters, while other types have even smaller limits.

In most computer applications a need exists to store much more data than can be stored economically in normal computer memories. In many of these cases the access time is relatively unimportant or can be minimized by special procedures. Magnetic tape offers a convenient means of storing data in such cases, since many millions of characters can be stored on one reel. Random-access times, however, are quite great—often in the order of minutes. The special procedures mentioned above for reducing time lost for access include the operation of storing data in some logical order on the tape. Processing occurs in this order as the tape is transported from one end to the other.

When more and more data need to be stored, extra reels can be used. In some applications these reels are changed manually on one tape unit; however, in many instances, multiple tape units have been successfully employed. In one inventory application of the Federal government, 100 tape units, all capable of independent operations, have been used to store the inventory records. In this application, a computer accepts input data, selects the proper tape unit, and reads the pertinent inventory record. The computer then combines the data to produce an updated result which is then recorded on the tape.

In some applications, the volume of data is not excessive, but the type of operations performed make magnetic tape admirably suited for the main storage. These operations involve sorting, collating, separating, selecting, etc. Systems have been developed consisting almost entirely of magnetic-tape units which are designed to do these operations. If the system is made up of four tape units, it can perform any one or all of these operations at the flip of a switch.

Input and Output. Devices used for input and output are inherently of a mechanical nature and hence are much slower than electronic computers. Frequently, therefore, when a great deal of input and/or output is needed for a particular problem, the speed of the computer cannot be fully utilized because of the slow terminal devices.

In order to conserve expensive computer time, input and output devices are often not connected directly to the computer. Instead, magnetic-tape units serve as a buffer between terminal devices, such as printers and keyboards, and the computer.

If a great deal of output is required, for example, a magnetic tape is connected to the computer and the output data recorded on the tape at the rate of a few thousand characters per second. At the completion of the run, the tape is removed to a tape reader which reads the information at a much slower rate and supplies it to a direct output device such as a printer.

The inverse process can also be accomplished. Equipment exists which will take information from a direct input device and record it on magnetic tape. The tape runs quite slowly in such instances, even stopping and starting under control of the input device. When controlled by the computer, the tape is then read at a high speed.

Equipment which is available for going to and from magnetic tape from direct terminal devices is described in Arts. 18.4.3 through 18.4.5.

18.4.3. Converters to Magnetic Tape. Equipment used for input consists principally of punched-card-to-magnetic-tape converters and punched-tape-to-magnetic-tape converters. Data from punched tape are not extensively converted because punched tape can often be read fast enough so that magnetic-tape buffers are not economically sound. The card-to-tape converters, on the other hand, are used extensively. The big advantage to this operation is that cards can be easily sorted into order before the data are recorded on tape. Relatively expensive tape sorters are not required in this case.

Some work has been done on keyboard-to-magnetic-tape converters. However, technical difficulties resulting in high costs have prevented large-scale acceptance of this method. In the future this may be a more important use of magnetic tape.

18.4.4. Converters from Magnetic Tape. Again, cards play an important role, receiving the data from tape with the use of tape-to-card converters. Of a great deal more importance, however, are magnetic-tape-to-printer converting devices. Line printers are particularly well matched to magnetic tape because the tape can

be transported at reasonable speeds. Still, magnetic-tape-to-typewriter converters have been on the market for some time and are extensively used.

Tapes produced by computers are used extensively for controlling other processes. Of coming importance in this respect is the controlling of machine tools and similar applications.

Requirements and Operational Details. The very fact that data are in discrete form allows digital tape units to be operated at varying speeds, as has been pointed out. This points up the principal difference in requirements between digital tape handlers and analog handlers, since the latter are run at constant speed. Indeed, the digital handler may be stopped and restarted in the middle of the tape without losing data if a short space is provided for stopping and gathering speed. Furthermore, the tape can be read in reverse because of the discrete character of the data.

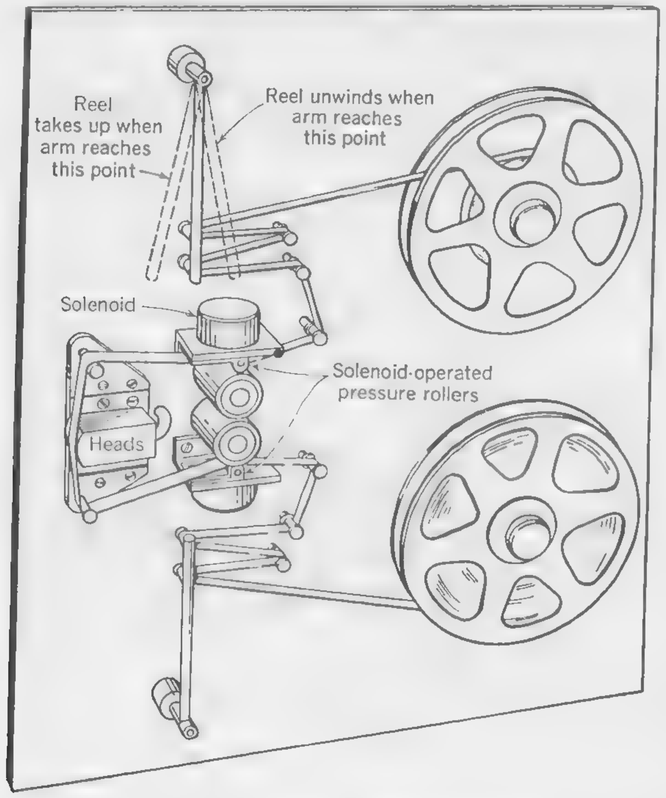


FIG. 18.16. A digital magnetic-tape unit.

The space provided for starting and stopping suggests the principal operational requirement of a digital tape handler; namely, the tape must be accelerated and decelerated in a short time to minimize the space that must be left between data points. The general standard for stop and start times of tape is in the vicinity of 3 to 5 msec for stopping from full speed and starting to full speed. To obtain these fast times, various manufacturers have used various techniques, a few of which will be described.

Figure 18.16 is a drawing of a tape handler similar to a type which is commercially available. Notice that the tape does not go directly from a reel to the head and drive unit. Instead, loops of tape about a tension arm provide slack between the reels and capstans (there are two capstans, one for each direction of tape travel).

As one of the capstans is energized (a solenoid-driven pressure roller presses the tape against the capstan), it need only accelerate the tape in the loops instead of overcoming the inertia of an entire reel of tape. Because the tension arms are pulling against each other, the capstan need not overcome tension-arm pressure. As the

tension arm moves, a point is reached (shown by dashed lines), at which a signal is sent to the reel motor to unwind tape. Hence the tension arm and the reel motor form a servo loop. Similarly, the take-up motor is servoed with the take-up tension arm.

Figure 18.17 shows another reel-servo-controlled unit which uses a vacuum column to keep tension on the tape slack left between reel and capstan. When the tape rises in the column to a certain point because of capstan actuation, a vacuum switch signals the reel motor to unwind tape; when the tape gets down beyond another point, the reel takes up tape.

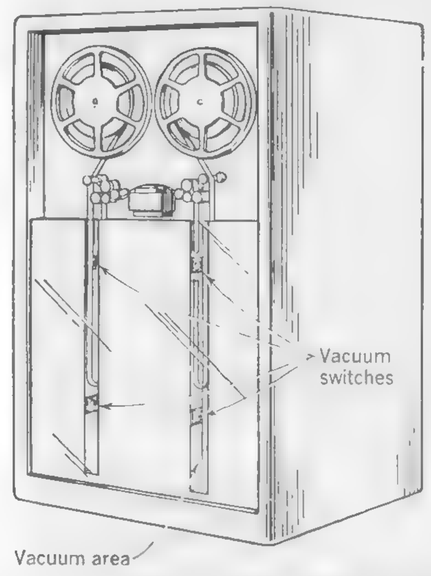


FIG. 18.17. Digital magnetic-tape unit with vacuum chambers.

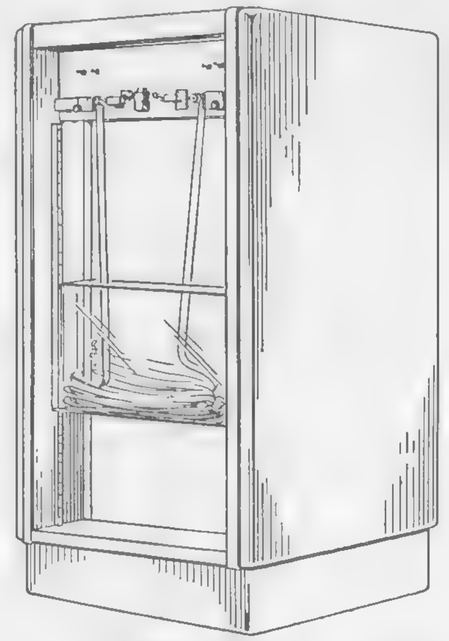


FIG. 18.18. Magnetic-tape unit with tape bins.

A technique first developed by the National Bureau of Standards lets the tape accumulate in and play out of tape bins, as shown in Fig. 18.18, instead of winding on reels. Similar units have been commercially available.

Speeds and Recording Density. Tape speeds vary from about 7 to 200 in. per sec, depending upon the frequency of the pulses being recorded. Pulse densities range from 50 to 700 pulses to the inch of tape; so that, when high recording frequencies are required, faster tape movement is used to keep pulse densities within the range mentioned.

Recording Techniques. Unlike magnetic-drum systems, most magnetic-tape read and write heads actually touch the tape as the tape is moved. Signal voltages obtained as output of the read head are usually in the range of a few millivolts (typical: 5 mv). Generally, a single set of heads is used for both reading and recording. Typical heads used for this purpose employ cores made of laminated-iron and nickel-alloy strips. Track widths are about 0.040 in. and from 6 to 14 heads per inch width of tape are typical. The number of windings varies widely with the application, but typical values are 100 to 300 turns. This gives inductances of less than 10 millihenrys and d-c resistances of 20 to 40 ohms. Record currents are 20 ma or less.

Characteristics of some recording methods are of interest. One method involves single-ended record amplifiers. This requires the degaussing of the tape with a-c signals, either with bulk erasers that demagnetize the entire reel of tape simultaneously, or with high-frequency a-c signals impressed on the head. The single-ended record currents then polarize the tape in one direction where a 1 is recorded; the tape

is left unmagnetized where a 0 is to be recorded. One disadvantage of this method is that rewriting on the tape cannot be made until the tape is erased.

Ordinary push-pull recording, in which the tape is magnetized in one direction for a 0 and in the other direction for a 1, is often used. An interesting variation of the push-pull recording technique takes advantage of the fact that the signal when read from the tape is a differentiation of the signal that was recorded. Figure 18.19 illustrates this technique. The magnetism has two directions of possible polarities, and the polarity is changed from bit to bit only when a 1 is to be recorded. Hence signals are obtained from the read head only when a 1 is recorded, and although successive 1's result in opposite-polarity signals, these signals can be rectified to provide a single-polarity wave.

18.4.5. Special Magnetic Input and Output Devices. The principles of magnetic recording are incorporated in special devices which do not employ "tape" in the normal sense of the word. These are described only briefly, as they are not as yet widely used.

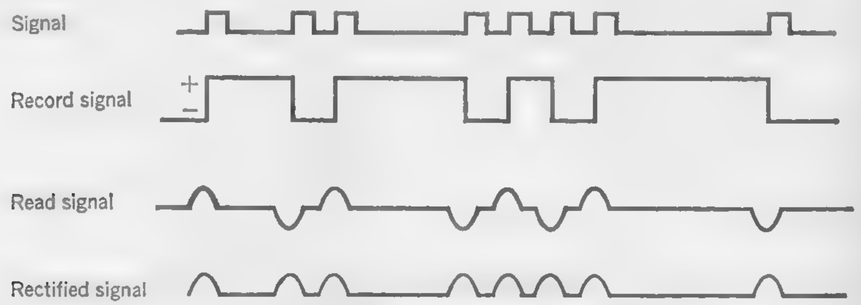


Fig. 18.19. Magnetic-tape signals.

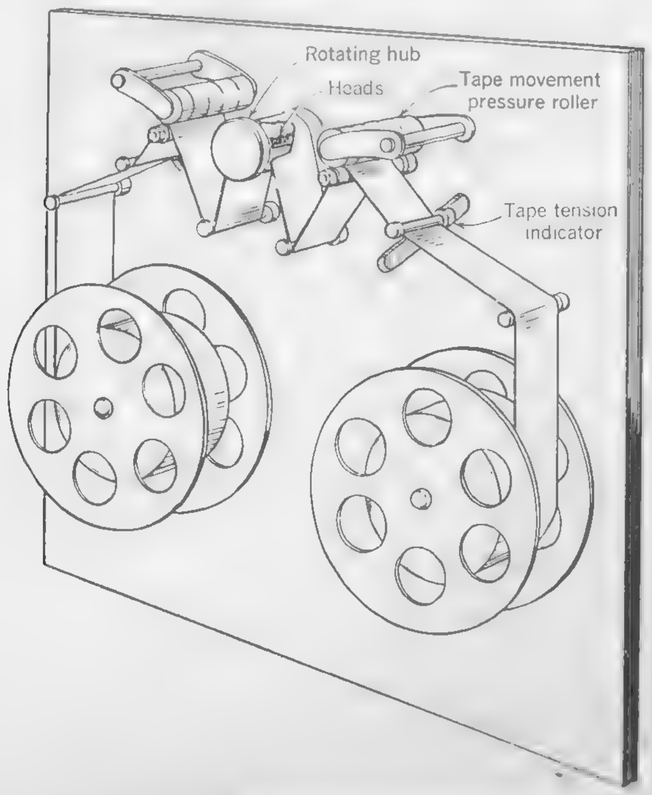


Fig. 18.20. Rotating-head tape system.

Rotating-head Devices. Figure 18.20 is a depiction of a device which does use a tape. The tape is moved endwise as before, but its movement is not necessary for the reading and recording of the information. Instead, the heads are mounted in a rotating hub and the information is read when the heads are rotated past the tape. This method necessitates the use of slip rings, and the hub is equipped with a fast-action clutch to obviate the necessity of continuous rotation.

Magnetic Page Writer. When used in the scheme presented in the preceding paragraph, the tape can be conveniently divided into parts, each part containing as much information as can be read by one rotation of the hub. These parts are often called "pages."

If, now, the hub is expanded in both length and diameter and more heads are added, the tape becomes very wide and can truly be said to resemble pages connected edge to edge. The movement of the tape selects the page to be read and the head or heads select the lines to be read on each page.

The hub, being quite large, is called a drum and must continually rotate at a few hundred rpm. To prevent wear, the tape does not actually rub upon the drum when not being read or recorded; the wind layer and centrifugal forces set up by the drum's rotation are sufficient to hold the tape away from the drum. When reading or writing is to be accomplished, the movement of rollers clamps and pulls the tape tightly against the drum (see Fig. 18.21). The tape, which is usually an endless strip stored in a bin, offers a convenient means of auxiliary storage.

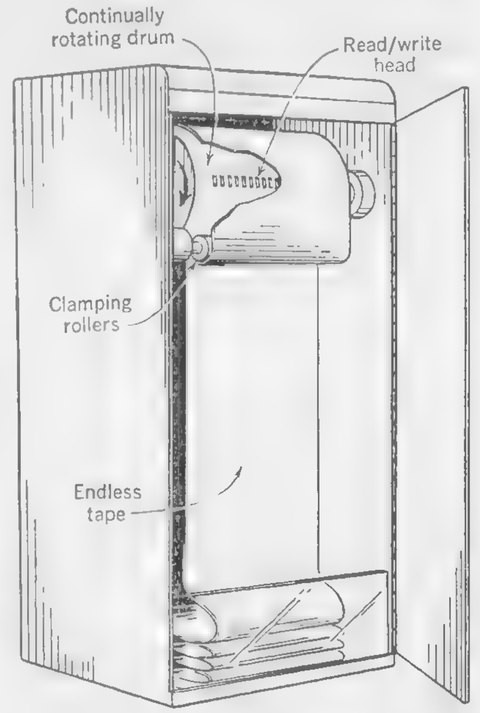


Fig. 18.21. Magnetic page writer.

Magnetic Cards. Pieces of paper or cardboard have been coated with magnetic dispersion so that information can be recorded on one or both sides. The normal method is to coat only one side so that information may be recorded on one side and the same information, or at least part of it, printed on the other side. These cards conceivably can be used much in the same way punched cards are used, although extensive use of them has not been done as yet. One advantage of magnetic cards over punched cards is that much more data can be recorded per card. Five hundred to a thousand alphanumeric characters can easily be recorded on one 3 by 5 card.

The mechanisms for handling the cards are the most intricate part of this equipment and are much the same as punched-card-handling equipment. Cards move at a speed of about 10 in. per sec.

An interesting application, for which equipment is available commercially, is that of up-dating bank statements. The normal statement that the customer eventually receives is a magnetic card with all the pertinent data recorded on one side as well as printed on the other, including the account balance. As each entry is made, the old balance is read from the card automatically, and the up-dating is accomplished by a computer using data entered from the keyboard. The new balance is then printed and recorded magnetically in place of the old balance. This eliminates the necessity for the operator to read the old balance and enter it manually.

Check Reading. Considerable effort has been devoted to printing of checks with magnetic ink to facilitate automatic reading. Equipment exists which will read check account numbers printed in coded magnetic ink. The most interesting aspect of this equipment is, again, its paper-handling feature. The paper checks must come

into contact with the read heads in order to be read; as a result, check mutilization is of primary concern. The equipment uses a pneumatic system in which a partial vacuum is used to transport the checks successfully past the read heads with negligible error in reading the information.

18.5. SYMBOL DISPLAYS*

Symbol displays are employed foremost for presenting data for immediate human consumption in contrast to printers, plotters, punched cards, etc., whose primary purpose is recording. Data should be displayed in a form which is readily interpretable by observation. Data inside the computer, though decipherable, are seldom immediately recognizable because of being coded in machine language.

The symbols most often used are the English alphabet and the arabic numerals plus a few others such as the period or radix point and + and − signs. The displays are normally a composite aggregate of these symbols with the computer in control of the symbol choice and sometimes even the display format. In some instances, the display is eminently involved in the machine's operation—for example, most desk calculators where cams, cogs, and gears on the display wheels control the arithmetic operations to some extent. In other cases the display exercises a form of control over the machine's operation. This usually takes the form of a time control wherein the read-out tells the machine it is set up and the machine then commences operation.

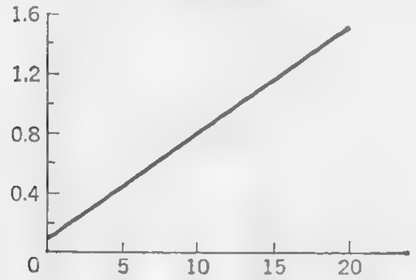


Fig. 18.22: Minimum symbol height (inches) vs. viewing distance (feet).

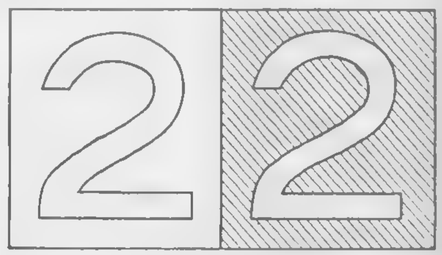


Fig. 18.23.

The machine may handle its information in either the serial or parallel mode. The information is best displayed when made available in its entirety or in symbol groups of reasonable size because the observer can more readily comprehend data presented in this manner than in smaller sequential groups.

The conversion from machine language to display code can be a major factor in over-all system design and to the designer is often of more importance than the appearance and possibly even the performance of the display. If the display code is chosen to be a "select one out of n" (one hot) code, nearly any display scheme can be used. But this is often expensive from the standpoint of equipment. Buffering equipment is normally used between the machine and the display. There are, at times, advantages to be gained by time sharing this buffer among the outputs and occasionally even among the inputs.

For maximum intelligibility the design of all symbols should be without flourishes. The critical details such as corners, openings, and intersections of the figures should be simple but prominent. Words should normally be oriented horizontally and should read from left to right. Other orientations lead to misreading of data. The spacing between symbols in a word should be approximately one stroke width, and space between related words on a single line should be approximately one symbol width. The height-to-width ratio for most symbols should be 3:2, though in the special case of drum displays a ratio of 1:1 is better. The stroke width of a black symbol on a white background should generally be about one-sixth of the symbol

* By William A. Farrand.

height; but with white on black one-seventh, and glowing symbols one-eighth, for the stroke tends to spread. In the English alphabet the capitals are more distinguishable than the lower-case letters.

The display is always more presentable when the symbols are formed in the plane of the bezel, eliminating parallax. This is especially true if the characters in a word are not formed coplanar, for the symbols then shift in position with respect to each other as the viewer shifts position. It is often of value to have the display intensity variable. This, of course, means general lighting as well as symbol-stroke illumination and is especially true in operational military equipment where night vision should not be hampered by the displays illumination.

The speed of operation and the mode of use will be discussed after the description of the various forms and types of displays.

18.5.1. Selectively Positioned Symbols. These are usually rotating indicators. In gear, bar, or string linkage machines, such as Babbage's, these displays did the job adequately. They have many characteristics which make them useful in today's machine, but here they are often remotely positioned rather than directly positioned units. The remote-positioning systems used for 360-deg indications are the various electromagnetic ones such as resolvers, synchros, magnesyns, and telegons; the coded combs, gears, or cams; and servo-driven units. Some rotate less than a full revolution and many of these make use of 100- or 240-deg D'Arsonval meter movements.

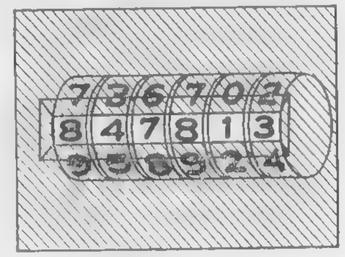


Fig. 18.24. Odometer.

Drum Displays. The most common device which functions as a symbol display is the number wheel used in the odometer of an automobile (see Fig. 18.24). It gives a digital representation of an analog quantity and, as such, provides a very compact presentation of high precision. In this use the over-all accuracy is far less than the precision due to inaccuracies of the metering wheel. In many desk calculators the only displays for intermediate and final results are wheel displays in which the wheels are designed in such a manner as to effect the computation.

The decimal counter wheels, as depicted in the odometer, constitute a highly accurate and precise device made from low-accuracy parts. They use intermittent gear carries with the backlash of no consequence as long as the sum of the backlashes in the carry position is less than ½ digit in all columns. A decimal device of n digits with the shaft on the least significant digit maintains full precision with a carry backlash of up to $\pm(18/n)$ deg at each station. Other than this, 1 part in 20 is sufficient accuracy to maintain full precision in the decimal counter. If used for following a slowly varying quantity, these counter wheels are excellent, for all the wheels are geared together and a single shaft is, therefore, all that needs to be controlled. If the number to be displayed does not vary at a low rate, the input shaft must go at a high rate and the entire unit becomes complex.

Independent drum displays are available without this disadvantage. They, in general, demand a complicated drive device if many symbols are utilized on a single drum. Another difficulty in the utilization of drum indicators is the large center-to-center distance necessary because of the thickness of the wheels mechanism and the large diameters necessary for a large character selection. On small drums a height-to-width ratio for the symbol of 1:1, instead of 3:2, is best because of the possible misplacement of the dial behind the bezel and the curvature of the dial.

Tapes.. These disadvantages are mainly offset by the use of tape for the display when a large group of characters must have one of its symbols selected. These tape units can be thought of as flexibly rimmed wheels which have most of the characteristics of wheels.

It is possible further to compress the size (a factor of 2) by making the tape into a Möbius band and having symbols printed on both sides; however, this will not allow one to transilluminate the tape. Another use for the Möbius band is to increase the space between character codes on the drive wheel by effectively interplexing the

symbols. Transillumination is applicable here. The code used for selection can be on the band itself, giving positive registry even with loosely coupled drive equipment.

Disks. Disks are also widely used for symbol displays. They possess many unique characteristics of form and shape, but in use they are almost identical to drums (Fig. 18.27).

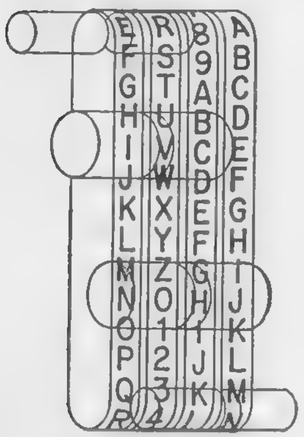


FIG. 18.25. A four-band alphanumeric display with staggered tape drives.

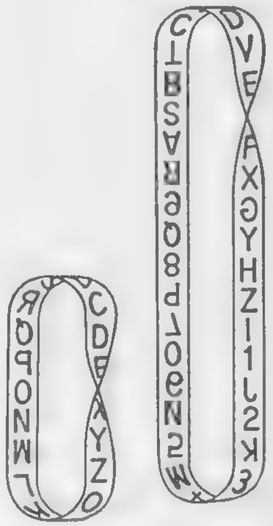


FIG. 18.26. Two Möbius bands—both containing alphanumeric data. The larger band reads consecutively every other character. The smaller band cannot be transilluminated, for different symbols are on opposite sides of the tape.

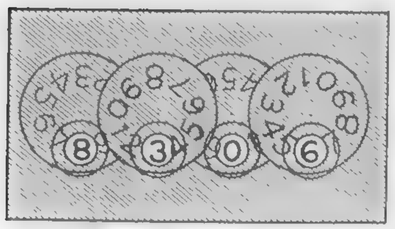


FIG. 18.27.

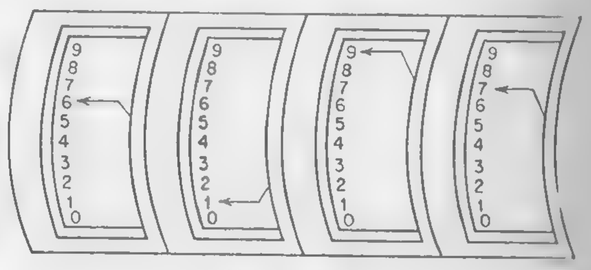


FIG. 18.28.

Pointers. One of the simplest read-outs is a galvanometer. The number is decoded by a weighted-resistance method into a current and this current is indicated on the meter. The meter scale is replaced with one containing the set of symbols corresponding to the coded currents; 90-, 100-, and 240-deg D'Arsonval meters are available with their pointers swinging in the plane of their faces, and edge-reading movements are available with their needles swinging orthogonal to their faces (Fig. 18.28).

This display leaves much to be desired from the standpoint of appearance and legibility, for the selected symbols must necessarily be separated by many times their symbol width. They are not well aligned for they are position-selected and the pointer may not indicate on the symbol center.

18.5.2. Selectively Illuminated Symbols. *Strobed Indicators.* Drums, tapes, and disks can be utilized by cyclically moving them in such a manner that the characters are displayed if illuminated. Then, once each cycle, the display is strobed at the time necessary to select and illuminate the intended symbol. This operation, as

exemplified in a very simple transilluminated decimal wheel rotating at nearly constant speed, appears in Fig. 18.29.

In order for a clean image to be viewed the light source should not be on for more than one-tenth of a character. This gives a duty cycle of one-hundredth, which even at 60 flashes per second is very fatiguing for the viewer. The image should be formed near the plane of the bezel so that motion of the viewer does not obscure or deteriorate the image.

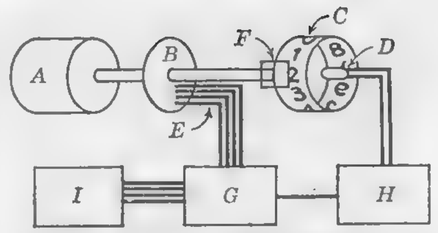


FIG. 18.29. A strobed indicator: *A*, motor; *B*, coded disk; *C*, drum; *D*, light; *E*, reading brushes; *F*, bezel; *G*, comparator; *H*, blocking oscillator; *I*, input register.

The flicker can be overcome by casting the image upon a phosphorescent material with sufficient lag to increase the duty cycle above the critical point.* This alone is not sufficient, for much of the light from the strobe passes through the phosphor without spectral frequency conversion so that even though there is phosphor afterglow the main intensity still is of low duty cycle. This is taken care of by a choice of phosphor and of illumination source, such that most of the light is of suitable frequency for exciting the phosphor, and then a light filter can be used to reject the strobe's light and to pass the phosphor's glow. This over-all technique is very simple to apply and yields excellent results.

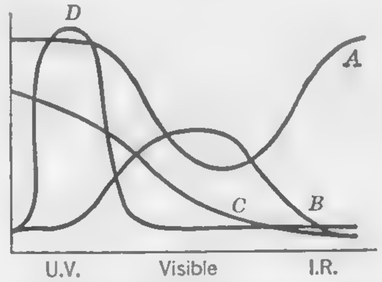


FIG. 18.30. *A*, filter-rejection curve; *B*, phosphor glow; *C*, phosphor sensitivity; *D*, light-source spectrum.

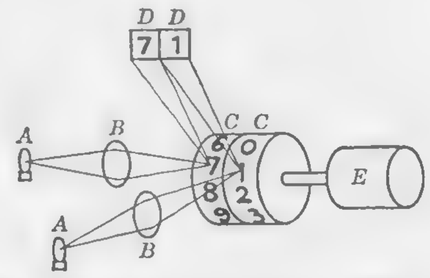


FIG. 18.31. *A*, light; *B*, lens; *C*, reflection-symbol drum; *D*, viewing bezel; *E*, motor.

FIG. 18.32.

In certain installations it is better if the data are projected to the viewing bezel, and in such cases, a simpler geometry and a higher efficiency are often available using reflection† rather than transmission.

Images with Fixed Register. Certain information can be symbolically displayed with clarity only if the various symbols are viewed in a given positional word structure. Displays which maintain this register irrespective of the particular symbol selected are discussed below.

Edge-lit Symbols. These normally take the form of a stack of sheets of clear plastic or glass on which the symbols are engraved, one symbol per sheet. By selectively illuminating the edges of these sheets the desired symbol, and it alone, can be forced to glow. This symbol is then viewed through all the preceding sheets and, as

* The critical point varies with frequency and is at 30 per cent duty cycle for 24 images per second.

† One of the best ultraviolet reflectors known is polished aluminum.

these sheets are translucent in the small area of their symbol shape, there is some intersymbol interference. To alleviate this, the symbols are sometimes shifted in reference to the bezel center and their stacking order changed. Another apparent shift of symbol position is due to parallax. This occurs when the observer's position is not in line with the symbols. The problem is emphasized by the images being formed in different planes.

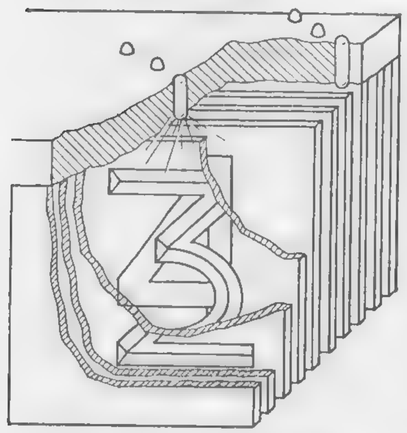


Fig. 18.33. Edge-lit display.

This form, one sheet per symbol, demands the minimum of decoding equipment. When a large set of symbols must be stacked it is often desirable from the standpoint of intersymbol interference and symbol misalignment to dissect the symbols so as to give fewer sheets in the stack. This requires a more complicated symbol-selection scheme (more than one wire per symbol and these wires are seldom symbolwise unique). Because of the low luminous efficiency, rather high intensity light sources such as filamentary bulbs are often resorted to, and this normally demands relay, thyratron, or magnetic-amplifier symbol switching.

Since these symbols are edge-lighted, care in the symbol shape should be emphasized in order to provide a uniform glow. A small amount of shape distortion (changes from the Futura or Groton bold fonts) often yields more legible characters. Care must be taken in the engraving of the symbols, as any tool loading or tool chatter yields an uneven symbol. If all symbols do not have the appropriate stroke width the display appears uneven, and because of the large commercial tolerances on plastic-sheet stock, care must be taken to set the graver depth from the engraved side and not from the engraver bed, as is standard practice.

Fig. 18.34. Symbol showing appropriate distortion for edge lighting.

Shaped-electrode Glow Tubes. These tubes consist normally of a single anode and a stack of individually symbol-shaped cathodes in a cold-cathode glow tube. The shaped cathodes are stacked one behind the other. These tubes have most of the properties of the edge-lit symbols and can be made in a similar multitude of configurations. Their high luminous efficiency makes direct electronic selection possible. Because of the small line size of the cathode and large size of the glow, there is very little intersymbol interference and therefore no need for symbol misalignment. Anode glow tubes can be made; they have similar characteristics. Tubes have been made in which the selected element is the cathode and all other elements act as the anode. These tubes must very definitely be fed d-c voltages of the proper polarity for if alternating current or the reverse polarity were applied all symbols would glow.

Selected Projector Display. Each image in the display can be formed by means of a group of simple projectors (one each per possible symbol) each of which is capable of projecting its image on the same screen and in the same register. The particular symbol to be displayed is selected by energizing its light source. These projectors can be extremely simple for a single symbol is the image, and if the lens systems contain distortion, the stencil can contain the complement of this distortion and the displayed image will be distortion-free.

Images with Functional Register. Where binary-coded decimal read-out is sufficient, positional notation similar to that shown in Fig. 18.36 is often useful. With the exception of radix point display nearly all these systems have as their main advantage simplicity of construction and decoding and have a major disadvantage of poor legibility. Glow transfer tubes and quantized beam-deflection tubes are examples of the extreme simplicity available in displays of this form.

Dot-formed Characters. Ever since the advent of the electric light bulb symbols have been displayed by the selective illumination of bulbs mounted in a planar array. An analysis of the alphanumeric characters shows that a 5 by 7 rectangular array of dots on constant center distance is sufficient to yield fairly recognizable symbols. As can be seen from Fig. 18.38, a few of these symbols, namely, 7, 8, *M*, *Q*, and *W*, leave much to be desired in their presentation. Many variations of the exact symbol shape are possible, especially if one is not limited to a constant center distance. Figure 18.39 shows the dot form of the numeric symbols and the + and − sign using a 3 by 6 array. It is to be noted that the horizontal and vertical connectivity is excellent, although the diagonal connectivity leaves much to be desired. If an *X* is used instead of a dot the diagonal connectivity is increased sufficiently to cause good symbol form with a decrease in horizontal and vertical connectivity so slight as to go unnoticed. The changing of the illuminators' shape from a dot to an *X* can be simply accomplished by masking the bulbs on the top, the bottom, and the two sides, leaving the four corners unmasked. By placing a diffuser (ground-glass screen) between the symbol and the viewer, the display can be made even more intelligible.

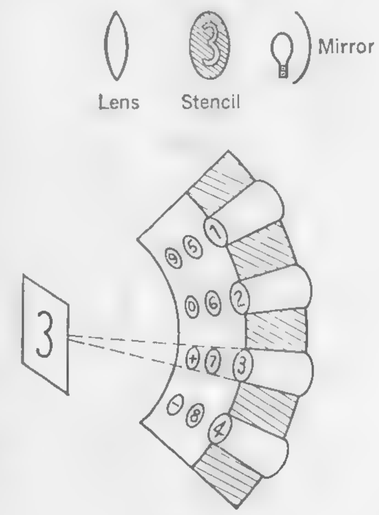


Fig. 18.35.

A simple selection matrix for the lights can be formed using one single secondary transformer per dot with multiple primaries. All dots which should be selected to display a single symbol can be threaded with one primary winding. This gives *n* primary threads for *n* symbols, but not all threads pass through all cores. If any one of these threads is a-c excited all the transformers which it links will cause their respective lights to glow. There will be no intersymbol interference due to coupling by the other primaries if the unchosen leads are left open-circuited.

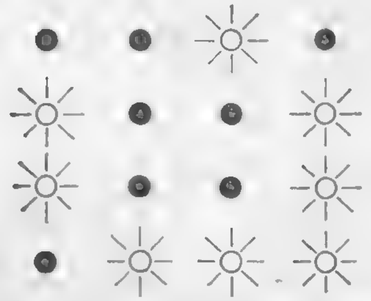


Fig. 18.36. A binary-coded decimal display of 6197.

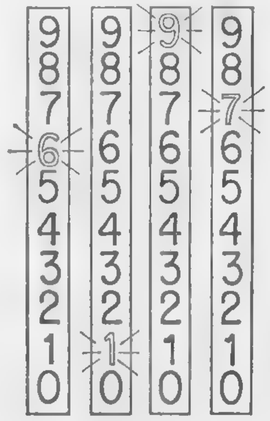


Fig. 18.37.

Special tubes are available which are admirably suited to this task in that a spot glow can be formed at any intersection of many horizontal and vertical wires by activating the appropriate single vertical and single horizontal wire. They can therefore be digitally selected. These tubes have the glow formed in such a manner that their face (a phosphor screen) glows and has considerable retentivity.

18.5.3. Cathode-ray Methods. Conventional tubes have been used to write as with a pencil and special tubes have been constructed which are able to write as with

a typewriter. Both have been used with holding guns to delete the rewriting process when the images are to be displayed for any length of time.

Conventional Cathode-ray Tubes. Symbols can be formed on the face of a cathode-ray tube if adequate control can be maintained over the electron beam. In a conventional cathode-ray tube control is available on intensity, focus, and position.

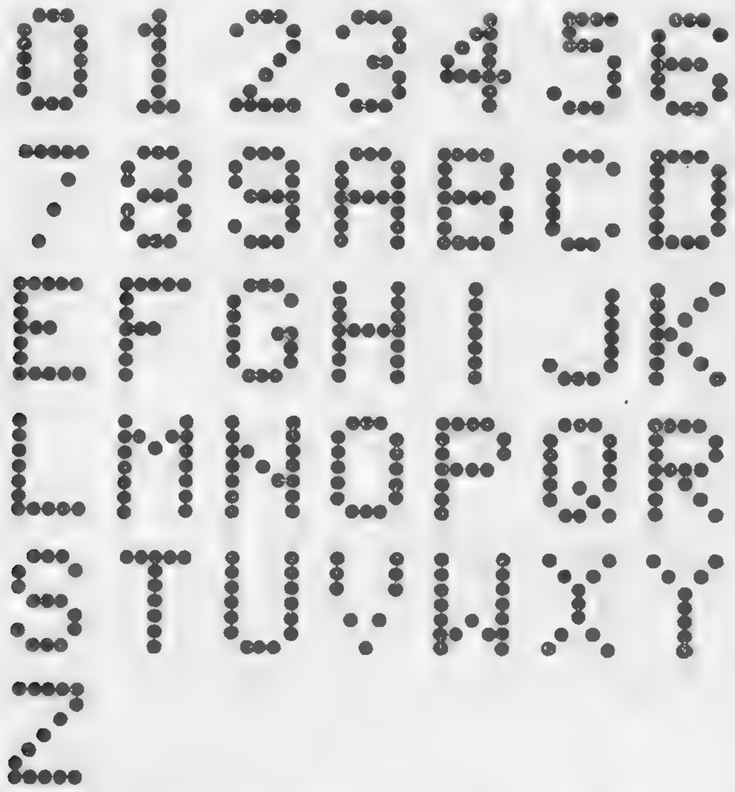


FIG. 18.38. Figures formed by a 5 × 7 array of dots.

FIG. 18.39. Numerals formed by a 5 × 7 array of dots.

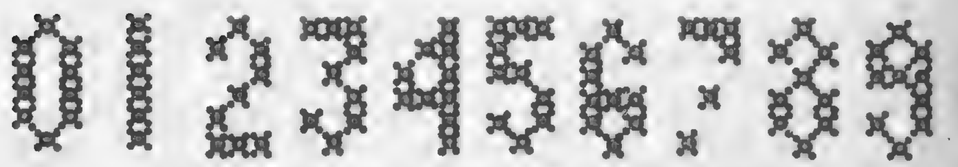


FIG. 18.40. Symbols formed with increased diagonal connectivity.

Various methods of controlling these in order to trace out the symbols are described below.

With the advent of the thin flat cathode-ray tube it is highly possible that this display form will take up the minimum of control-panel space and weight. Control applications exist where this is extremely important. In some of these tubes digitally selected deflection is used. This increases their electrical compatibility with the computer.

Monoscopes. Monoscopes are similar to iconoscopes in construction but in place of the photocathode there is a fixed pattern on an oxide coating. The secondary emission of the printed portions is low while that of the oxide is high, giving rise to a variable secondary electron current to the collector as the imprinted cathode is scanned. If one cathode-ray tube were used per symbol and an ensemble of monoscopes were available (one per character) from which the symbol selection could be made, the symbol selection and format control would be obvious.

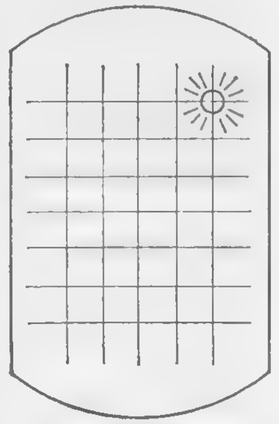


FIG. 18.41. Lattice glow tube.

The amount of equipment in the display can be decreased in two ways at the expense of control complexity. The first is the use of a single monoscope containing all characters. The sweep would then be a raster scan just large enough to cover one symbol. A second deflection potential (applied to the same deflection plates) would select the appropriate symbol from the array for interrogation by the raster-scanning beam. The second way is the use of a single presentation tube with a sweep in synchronism with the monoscope's raster scan and of such a magnitude as to give the appropriate size and shape to the displayed symbol. The location of the selected symbol on the face of the viewing tube can be controlled through the deflection circuits that control the scan.

Dot-formed Characters. If the information concerning the dots which form a given character is available, then it can be sequentially sampled; and if the beam of a cathode-ray tube is swept in such a manner that its position at the instant of a given dot sample is at that dot's position (and the beam is either intensified or attenuated according to the value 1 or 0 of the sample), then the symbol will be dot-formed on the face of the cathode-ray tube. The position of the character on the face of the tube can be controlled by means of the deflection circuitry for the tube, as can the size and shape of the image. Suppose that a memory system contained the dot information for all the characters which might possibly be selected and that the dots were time-sequentially scanned. A simple memory for this consists of an array of switch cores (one per dot) which can be used with circuitry which sequentially sets and then resets each core in turn. If a reading thread now passes through only those cores corresponding to the dots that are to be made luminous for a specific symbol, the signal out on that line will correspond to the desired intensity signal as the cores are interrogated. A core array is similar to the transformer array discussed under Dot-formed Characters. One core array will suffice for any number of characters.

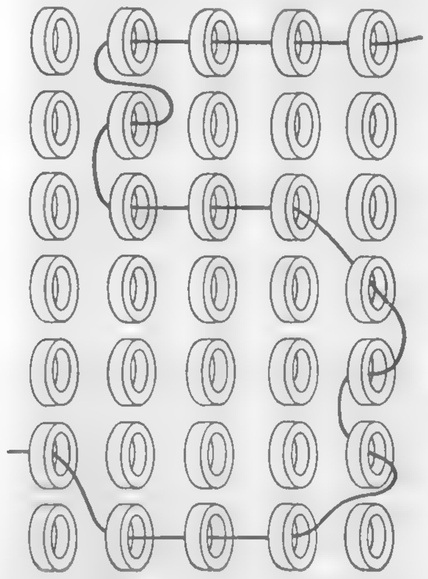


FIG. 18.42. An array of switch cores for the dot generation of symbols (a read thread for the symbol "5" is shown).

A magnetic drum can also be used effectively as the memory for the dot-formed characters, and its addressing and information sequencing can control the sweep of the cathode-ray tube, producing the symbols on the face of the tube. This technique can easily be exploited by utilizing other memories.

Deflection-formed Symbols. Lissajous figures can be utilized to produce symbols for a display. However, to do this with sine waves would demand many harmonics and accurate phase and amplitude control. Since most symbols can be thought of as

combinations of straight lines and segments of ellipses, segmented sine waves and d-c values can be combined with proper blanking control to give clean-cut images with very little circuitry. Figure 18.43 shows the form of the symbols +, −, and 0 through 9; while Fig. 18.44 lists the functions which will adequately produce them.

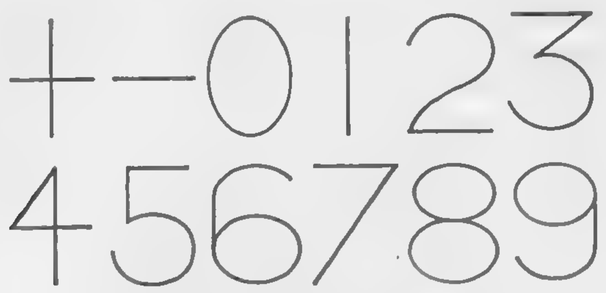


FIG. 18.43. Deflection-formed symbols.

Symbol	e_x	e_y	*Blanking*
0	$A \sin 2wt$	$B \cos 2wt$	None
1	0	$B \cos wt$	None
2	$A \sin 2wt$	$-\frac{3}{4}B + \frac{9}{4}B \sin wt$	225 to 315 deg
3	$A \sin 2wt$	$B/2 - B/2 \cos 2wt;\ 0 < \theta < 180$ $-B/2 + B/2 \cos 2wt;\ 180 < \theta < 360$	135 to 180 deg 315 to 360 deg
4	$A \sin 2wt;\ 0 < \theta < 180$ $0;\ 180 < \theta < 360$	$0;\ 0 < \theta < 135$ $B + B \sin 2wt;\ 135 < \theta < 225$ $+B \sin 2wt;\ 225 < \theta < 360$	None
5	$-B/2;\ 292 < \theta < 337$ $A \sin 2wt$; elsewhere	$-B/3 + 2B/3 \cos 2wt;\ 90 < \theta < 270$ $+B \sin 2wt;\ 270 < \theta < 337$ $B;\ 337 < \theta < 450$	90 to 135 deg
6	$A \sin 2wt$	$-B \cos 2wt;\ 0 < \theta < 180$ $-B/2 - B/2 \cos 2wt;\ 180 < \theta < 360$	22 to 45 deg
7	$.293A$ − F.W. Rect. $2A \sin wt$	$.586B - \frac{1}{2}$W. Rect. $2B \sin wt$	None
8	$A \sin 2wt$	$B/2 + B/2 \cos 2wt;\ 0 < \theta < 180$ $-B/2 - B/2 \cos 2wt;\ 180 < \theta < 360$	None
9	$A \sin 2wt$	$B/2 + B/2 \cos 2wt;\ 0 < \theta < 180$ $B \cos 2wt;\ 180 < \theta < 360$	225 to 360 deg

FIG. 18.44. Deflection functions for symbol presentation.

It should be noted that some of the symbols can be very easily formed by this technique (e.g., the figure 7 which is a full-wave-rectified sinuosoid beat against a half-wave-rectified sinuosoid). The circuitry involved in display of the numerals becomes considerably simpler when a height-to-width ratio of 2:1 is used (e.g., 8 is given by $A \sin wt$ vs. $B \sin 2wt$ and 3 is the same as 8 with appropriate blanking). The blanking is less critical for 2:1.

Image-converter Tubes. This type of tube is most commonly remembered as used in the sniperscope of World War II fame. In that form it consists of a monomolecular photocathode sensitive to infrared radiation, a deflection system and electron optics which control and focus the image formed on the photocathode to the phosphor anode in such a manner that the image is visible. In a form suitable for use in a symbol display the photocathode would be replaced, as in the monoscope, by a fixed pattern on an oxide coating with the secondary emission of the printed portions very low while that of the oxide would be very high.

If a flood of low-velocity electrons were to illuminate the image plate (secondary cathode), the image could be made visible on the phosphor anode. An extremely useful device can be made by having all desired characters printed on the cathode, with selection accomplished by means of the flood-gun deflection and format con-

trolled by the tube's deflection system. This system is then effectively a monoscope and a viewing tube all in one.

Shaped-beam Display Tubes. Special cathode-ray tubes have been constructed in which an electron beam of considerable cross section can be selectively deflected in such a manner as to illuminate any symbol in the stencil matrix. The stencil shapes the cross section of the beam to the form of the selected symbol. This beam is then deflected by a second deflection system to its prescribed place on the face of the tube where the phosphor changes the electron image into a luminous one. Most designs attempt to bring the beam back on axis before the second deflection plates (the image-positioning plates) act upon it. If it is not brought completely back on axis then

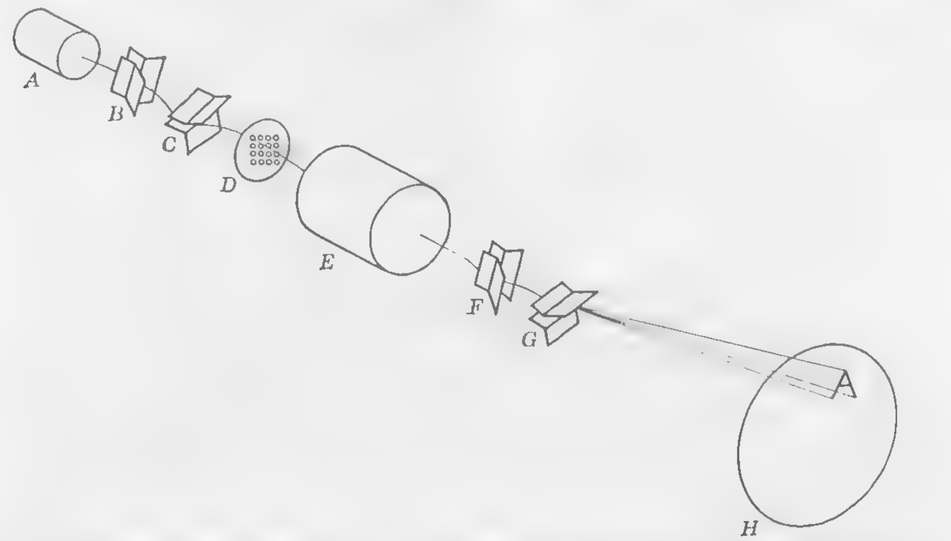


FIG. 18.45. A shaped-beam display tube: *A*, electron gun; *B* and *C*, symbol-selection plates; *D*, symbol-stencil matrix; *E*, electron lens; *F* and *G*, image-positioning plates; *H*, viewing screen.

some of the signal applied to the first deflection plates (the symbol-selection plates) can be subtracted from the format signal applied to the second, thus effecting reasonable control over the symbol placement. This axial alignment is accomplished by means of the second set of electron optics, which is conventionally a magnetic system at least in part.

Shaped-beam display tubes are especially useful as the P.P.I. (Plan Position Indication) scope in surveillance radar work for they readily allow identification markers to be displayed by the side of or in place of bogies. The radar map is traced by using a small hole, usually the central symbol in the stencil matrix, and the sweep and intensity controls are used as in conventional P.P.I. tubes.

A hold gun can be used in the tube to spray low-velocity electrons onto the screen and maintain the potentials of the printed pattern if the screen has appropriate secondary-emission characteristics. Normally these tubes do not have provision for selective erase as erasing is accomplished by disabling the flood gun.

These tubes have been made in sizes ranging from 3 to 19 in. Because of the complicated electron optical structure and the difficulties of folding it these tubes are all quite long.

18.6. ANALOG-TO-DIGITAL CONVERTERS*

These devices convert from analog magnitude to digital value. The analog variables sometimes interpreted are from a large class of which a few are acceleration, altitude, capacitance, current, energy, fluid flow, light intensity, mass, orientation,

* By William A. Farrand.

repetition rate, shaft position, temperature, time, voltage, and volume. When computation, documentation, or storage of these magnitudes is needed the conversion to digital value is often necessary. Some converters are especially useful in measuring quasi-static variables (upon command of the machine or its operator) rather than dynamic data. All converters are essentially counting, weighing, or reading devices. Converters can be incremental or value-measuring devices. If operating as value-measuring devices they are either value-reading or value-searching.

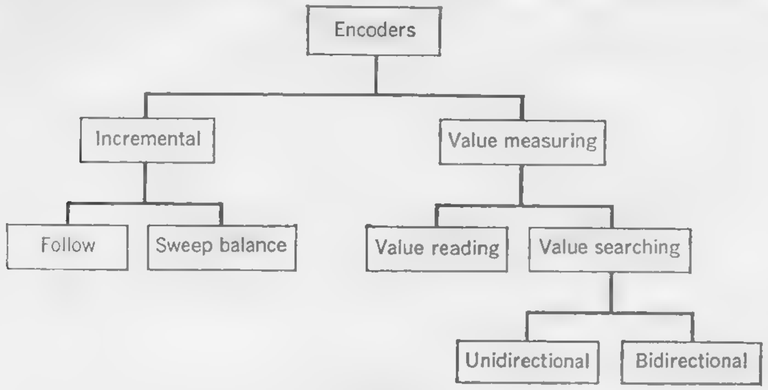


FIG. 18.46.

If value-searching they search in either a unidirectional or bidirectional manner and they initiate their search either at the last value of the function or at a predetermined value.*

The precision, stability, accuracy, resolution, sampling time, and sampling rate are extremely important properties of each converter. Sampling-time accuracy, value accuracy,† and sampling rate completely denote the functional distortion due to the digitizing operation. In sweep-balance converters and some others the sampling instant is not a priori defined for it is a function of the value of the variable. The output from the converter should be compatible with the machine that is to accept and operate upon its digital data. The code form of the output should therefore be essentially that of the machine.

18.6.1. Voltage-to-number Converters. In instrumentation, the transformation of measured variables to voltage is common because of the general simplicity of the equipment and the ease of distributing the voltage. This makes a voltage-to-number converter quite compatible with present-day process-control systems, for the instrumentation already in existence is often of this form. When the voltage does not vary greatly during the sampling interval accurate encoding can be performed with only a power amplifier between the encoder and its input. Often, however, the input voltage is not so limited and it is necessary to form a second voltage equal to the input at a given instant (the sampling instant) and to maintain this voltage even under load until the next sampling instant. There are applications in which it is not of consequence that the input has varied appreciably during the sampling interval and clamps are therefore not necessary. When clamps are used in high-accuracy converters they generally contribute error, especially when setting time must be short and clamping interval long. They have errors in amplitude due to stability, load regulation, and sensitivity and errors in instant of sampling due to timing and frequency response.

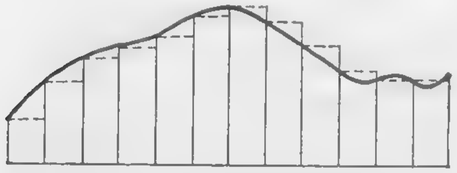


FIG. 18.47. A function sampled and clamped over the sampling interval.

* This is usually the mean value, the least upper bound, or the greatest lower bound.

† The value accuracy is dependent upon both the quantum and the accuracy of the bounds of the quantum.

Ramp Compare. These devices are truly voltage-to-time-to-number converters and utilize the coincidence of the input voltage with a voltage-time ramp as the event that is timed. It is conventional to enable a gate between a clock and a counter when the ramp starts or crosses datum and to disable this gate when the input voltage is of the same value as the ramp. The precision of this device is with respect to full scale and is a function of clock rate divided by ramp-repetition period. The stability is a function of clock stability voltage, comparator stability, and the stability of the

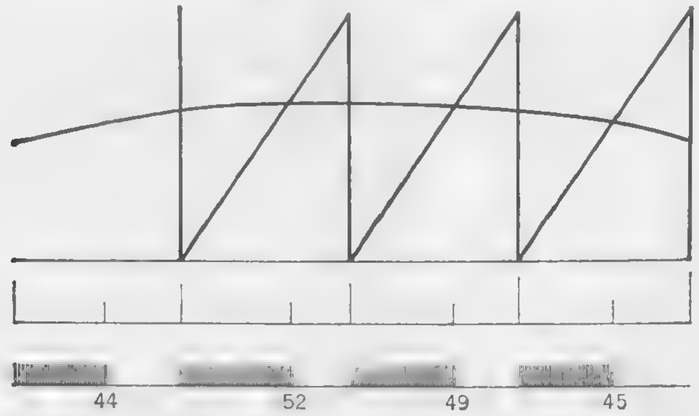


FIG. 18.48. Ramp compare converter waveforms.

ramp function itself. The accuracy is a function of the accuracy of the above-mentioned three items. The resolution is closely allied to the precision but is also a function of the short-time stability, for the resolution is the least count provided it is immediately repeatable. The sampling rate is determined by the repetition rate of the ramp, and the sampling instant is clearly a function of the value of the variable unless the sample is clamped for the measuring interval. The actual sampling instant is a function of the voltage. In Fig. 18.49 it can be seen that the actual sampling instant $t_i = t_o + nT + t$ where $t = eT/E$ where e is the voltage at t_i.

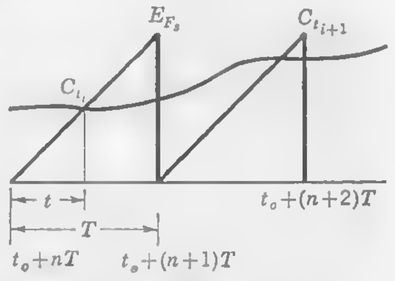


FIG. 18.49.

The sweeps are commonly, though not necessarily, linear, and when the over-all accuracy is to be no better than 0.05 per cent and the sampling rate is to be in kilocycles a phantastron ramp generator is often used. For a lower accuracy (approximately ½ per cent) constant-current pentode relaxation oscillators are often used. For even less accuracy (approximately 5 per cent) delay lines or even simple capacitor-charging devices are used. For higher accuracy special mechanical potentiometer circuits seem to be necessary, with a resultant lowering in the sampling rate to below a thousand sweeps per second. The d-c trigger pair and the multiar are the two most common voltage comparators; however, when more sensitivity is desired stabilized amplifiers often precede these voltage comparators.

The transient response of the comparator circuitry affects the accuracy of the values at the beginning and at the end of the sweep much more than it does the values in the central portion. This can be seen from a Fourier approximation to the ramp in which the higher frequencies are attenuated more than the lows. In limited-band-pass systems this problem can be alleviated by having the range of the ramp larger than the range of the voltage to be encoded. The zero-crossing detector can usually be made more sensitive and more stable if it is measuring one voltage with respect to ground instead of one voltage with respect to another. This can be done if the input and the ramp are subtracted and their difference is compared with zero.

Cathode-ray Tube and Mask. Figure 18.50 shows a mask which when illuminated by a sheet beam of narrow width illuminates the receptors in the normal binary code for the value of the sheet's position. This output is in parallel form; however, if all the receptors were tied together and the beam (now a pencil beam) were to be positioned as before and then swept across the sheet the output would be the serial form of the normal binary code for the beam's position. The output could either be least significant or most significant digit first, depending on whether it was swept up or down.

The cathode-ray tube is a convenient means of forming the beam for either serial or parallel operation. The mask is either inside or outside the cathode-ray tube. When inside, electron illumination is used and simple wires are the receptors. When the mask is exterior to the tube photon illumination is used and photosensitive elements are the receptors. When the mask is inside the tube it is possible to quantize the signal (that is, cause it to not be halfway between two discrete values) by means of a grid structure which forms a knife-edge electron lens. When the mask is exterior to the tube and a nonprogressive code is being used, quantization can be formed by an auxiliary circuit consisting of windows in the mask outside the code area and a photocell together with circuitry which servos the beam always to the center of one of these windows. This output, if not cyclic-coded, contains carry ambiguities and therefore reading is often inhibited while carries are being processed. The

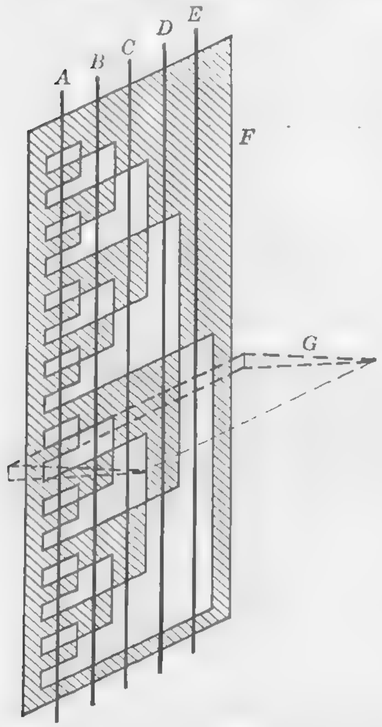


FIG. 18.50. 5-bit mask with sheet-beam illumination: *A*,*B*,*C*,*D*, and *E*, receptors; *F*, coded stencil; *G*, sheet-beam illumination.

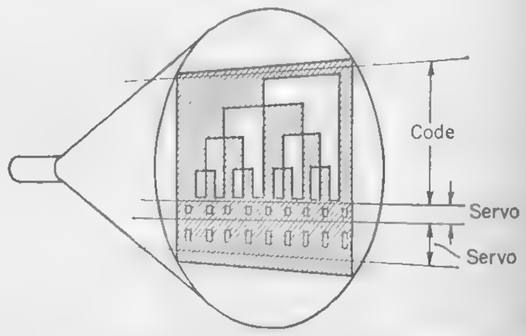


FIG. 18.51. Cathode-ray tube and external mask for photoelectric reading showing two channels for servo quantizing.

normal accuracy of these devices is 0.5 per cent with 0.1 per cent being almost the maximum obtainable because of the nonlinearities in the cathode-ray tube. Mirrored galvanometers are sometimes used in place of the cathode-ray tubes.

Voltage-to-shaft Rotation-to-number Converters. These normally consist of a voltage-to-shaft position servo together with a shaft position-to-number converter. The shaft position-to-number converters and the voltage-to-shaft position servos are discussed elsewhere. These units are characterized by high precision and slow speed and are best used to follow a function rather than to establish a value.

Digital-potentiometer Servos. The digital potentiometers are described later. As their title indicates, they are number-to-voltage converters, and therefore, when used in the feedback of a servo, form voltage-to-number converters.

Selective Subtraction. These methods involve the subtraction of weighted voltages from the unknown until the resultant is negative, then the removal of the last weighted voltage and the continued subtraction until the resultant is again negative and its

removal, etc. As an example, consider any positive nonfractional decimal number of n digits. This can be expressed as

$$a_n 10^{n-1} + a_{n-1}10^{n-2} + a_{n-2}10^{n-3} + \cdots + a_2 10^1 + a_1 = a_n, a_{n-1}, a_{n-2}, \ldots, a_2, a_1$$

The values of the a's are limited to the integers 0 through 9. The problem is now to determine the coefficients. First, assume $a_n = 1$ and try this by subtracting a voltage corresponding to 10^{n-1} from the voltage being converted. If the resultant voltage is negative then a_n is made zero and the subtracted voltage is removed. If, however, the resultant is greater than or equal to zero, another 10^{n-1} worth of voltage is either subtracted, or the first subtracted voltage is removed and a voltage corresponding to $2 \times 10^{n-1}$ is subtracted. The value of a_n is determined by successively trying all values 1 through 9 and testing until one of them produces a resultant voltage that is negative; then a_n is decreased in value by 1 and its corresponding voltage deleted, or if none produces a negative resultant the value is 9. a_{n-1} is now assumed to be 1 and the process is repeated for all values of all a's until the correct one for a_1 is determined. This process yields a decimal number in serial form, most significant

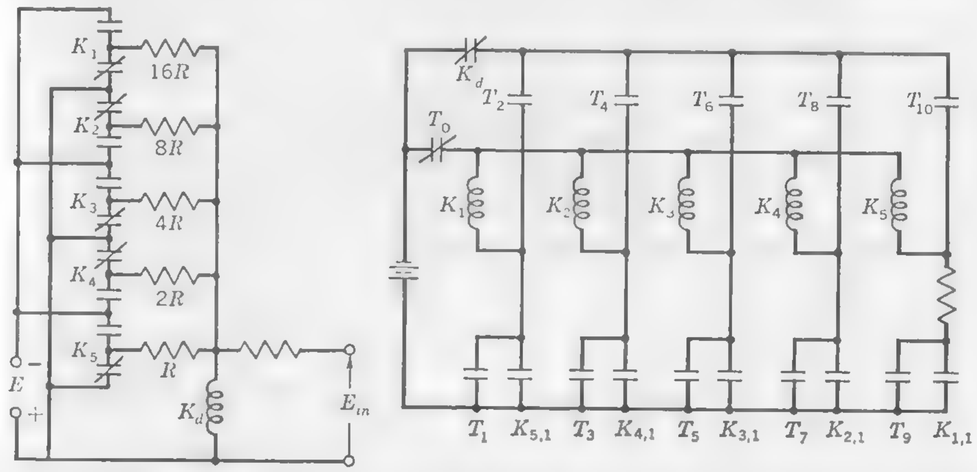


FIG. 18.52. Relay binary subtractive potentiometer servo. K_1, K_2, K_3, K_4, and K_5 correspond to the bits of the binary number where K_1 represents the least significant bit, K_2 the next least, etc. K_d detects the value of the summing point and the T's are from a timing stepping switch.

digit first. This form of number delivery is useful for transforming the number to parallel or to simple cyclic code but is rather cumbersome for normal serial work which needs the least significant digit first.

The above process can be modified to encode a voltage into any radix-numbers system. The device which determines whether the result is negative or not must have a stability and sensitivity equal to the precision expected.

In the above description the assumption is made that all the numbers are positive and nonfractional. Actually, fractions fit the theory perfectly and the methods therefore apply. The mathematical expressions shown were chosen to promote clarity in the explanation. Negative voltages can be made positive by subtracting a full-scale negative voltage from them and then adding its value numerically to the results. Another method is to detect the sign initially and, if negative, prime all the operations stated in the description.

Selectively Add-subtract (Binary). It can be easily demonstrated that $a_n, a_{n-1}, a_{n-2}, \ldots, a_2, a_1, -2^{n-1} \geq 0$ if and only if a_n is 1, and conversely it is negative if and only if a_n is 0.

This characteristic can be used to generate the binary number corresponding to a voltage. First, subtract a voltage corresponding to 2^{n-1} from the voltage being converted. If the resultant voltage is greater than or equal to zero, the value 1 is

taken as a_n and a voltage corresponding to 2^{n-2} is subtracted; but if the resultant is negative, the value 0 is taken as a_n and a voltage corresponding to 2^{n-2} is added. If this resultant is greater than or equal to 1, the value 1 is taken as a_{n-1} and a voltage corresponding to 2^{n-3} subtracted, etc. This process is continued in sequence until a_1 is determined; it produces the binary number in serial form, most significant digit first.

In comparing this scheme with the preceding binary subtraction method it should be noted that two power supplies are necessary rather than just one, for voltages must be both added and subtracted. The selective subtractive method costs two

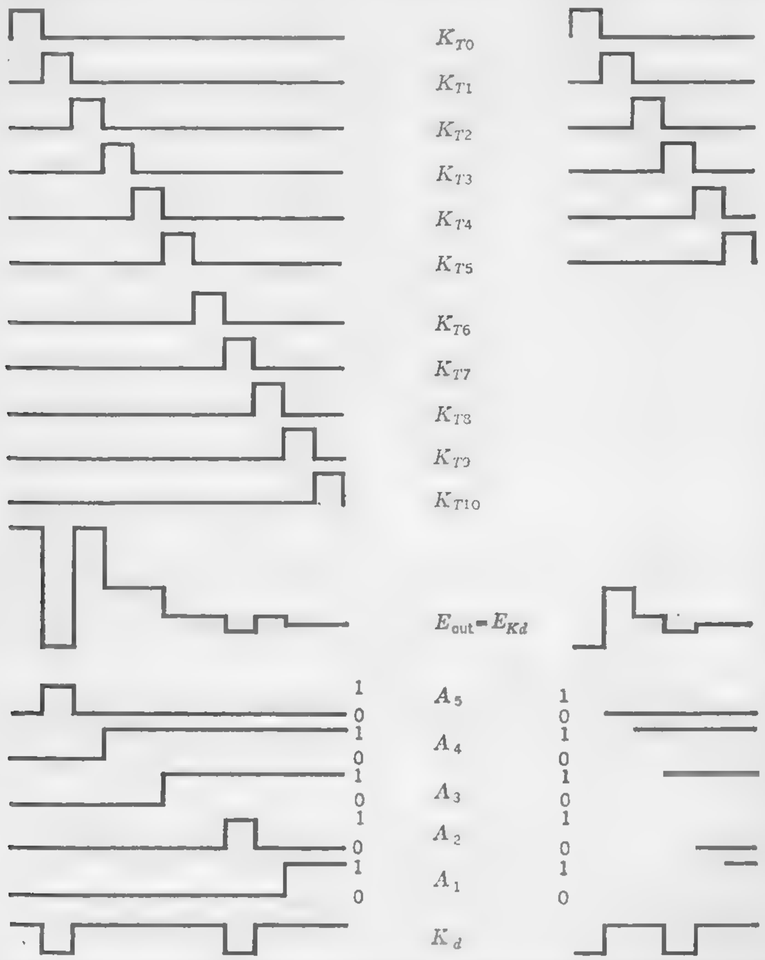


FIG. 18.53. Sync timing plot for binary subtraction and binary add-subtract relay servos.

operations (trial and possible retraction) per bit, while the selective add-subtract method uses only one operation per bit. Therefore the selective add-subtract method is synchronizable at maximum clock rate, and this is of value when computation can be performed on this most-significant-digit first serially presented number.

Add-subtract Counters. If a register is interconnected with logic so that it can operate as an add-subtract counter and control a weighted resistor matrix so as to produce a current or voltage proportional to the number stored in the register then this can be servoed by gating in clock pulses on the add or subtract bus at the control of a comparison circuit. The comparison circuit causes the counter to add clock pulses when the register's decoded voltage is lower than the input voltage, and to subtract when above. This comparator must have a dead zone in the vicinity of zero or the counter will be caused to hunt at clock frequency for a static input.

Capacitor Discharge to Least Significant Digit. This is a voltage-to-time-to-number converter in which the voltage-to-time relationship is not linear. As can be seen in Figs. 18.55 and 18.56 if the voltage to be measured is placed on a capacitor and this capacitor is discharged through a resistor R the time to discharge to the least significant digit (one unit) will be $RC \ln E/E_u$. If instead of a fixed resistor a constant-current load is applied, then a linear, not a logarithmic relationship, would hold.

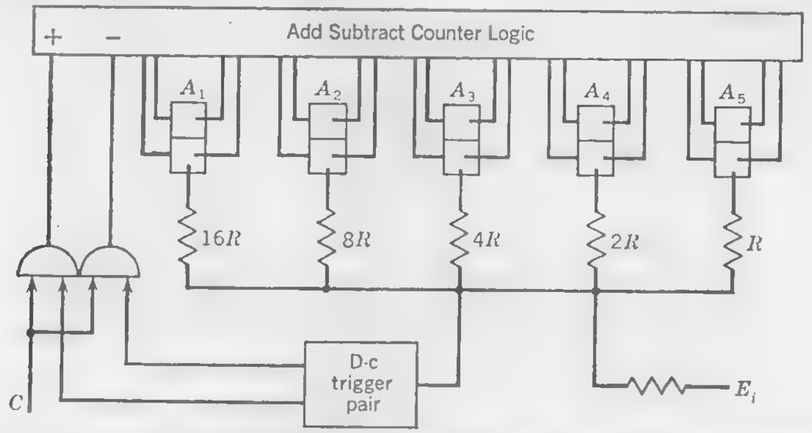


FIG. 18.54. Add-subtract counter servo potentiometer circuit utilizing flip-flops.

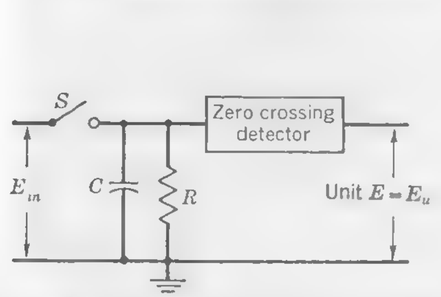


FIG. 18.55. Capacitor discharge to least-significant-digit circuit.

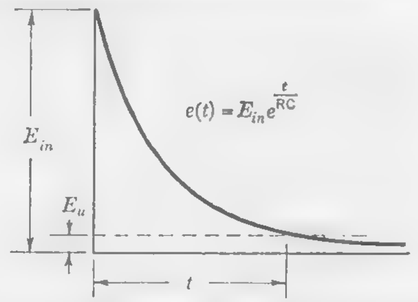


FIG. 18.56. Discharge waveforms. E_u unit value of voltage; t times to the least significant digit.

The time t can be quantized and evaluated by any of the means described in Art. 18.6.3. However, the most common method is that of counting clock pulses from start of discharge to coincidence with least-significant-digit value.

Other means for controlling the decay can be applied, and these will give other relationships between the number and the voltage.

18.6.2. Position-to-number Converters. For the convenience of discussion, position will be considered to be angular position, and for the instances where this is not correct, two recourses may be considered. First, the position might be transformed to angular position by racks and pinions or other linkages; second, the various converters described can also be constructed as the layout developments (planar representations) of the circular units described.

The feature of shaft position-to-number converters which seems to control their forms, complexity, and utilizability is the suppression of carry ambiguities. Each time that there is a shift in the value of any digit other than the least significant one, there is also a change in the least significant digit if its code is that of a radix-number system. If special care is not taken it will be possible to sample this output during this carry time and give an erroneous answer.

Incremental Techniques. The simplest way of digitizing a shaft position is perhaps to count the number of revolutions or segments of revolution that a shaft makes past

some fixed reference point. If the shaft rotates a sufficient number of times or is divided into sufficiently small countable segments, almost any accuracy can be obtained. A shift of datum due to a loss of pulses is a shift that remains with the converter until the datum is reestablished. It is normally necessary that the counter which is keeping track of the counted segments be an add-subtract one and further that there be intelligence available concerning the direction of rotation of the shaft. It is obviously necessary that the code for this be totally unambiguous, for it is within reason to assume that the shaft could rock to and fro across a count station and if ambiguity were present it might rack up a large and erroneous count.

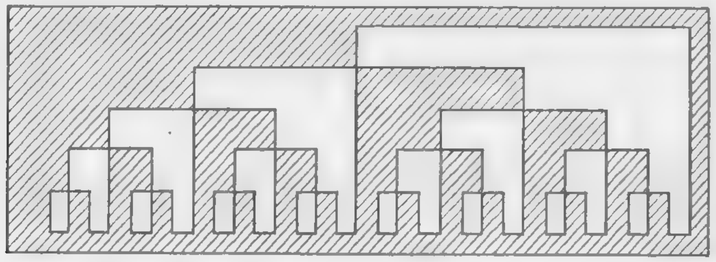


FIG. 18.57. Layout development of 5-bit binary wheel.

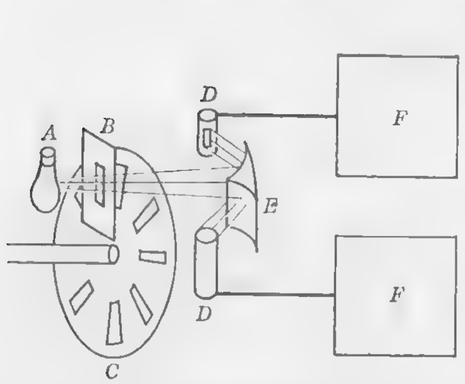


FIG. 18.58. Photoelectric shaft sector counter: A, light; B, slit mask; C, sector wheel; D, photoelectric cell; E, beam-splitting mirror; F, d-c trigger pairs.

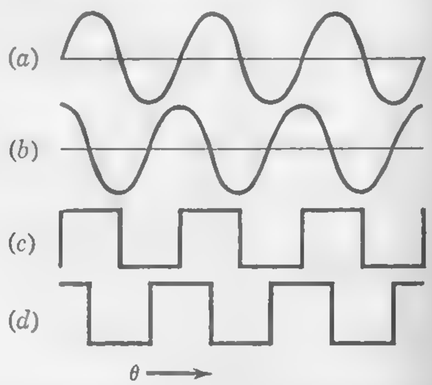


FIG. 18.59. Sector-detector waveforms: (*a*) one photoelectric-cell voltage; (*b*) other photoelectric-cell voltage; (*c*) d-c trigger-pair output for A input; (*d*) d-c trigger-pair output for B input; θ shaft rotation.

The output from a single sector detector can often be assumed to be sinusoidal. If a second sector detector is placed to read the same sector as the first one, but spatially displaced from it so that its output is spatially phase-shifted from it by 90 deg, the output can have its direction of rotation interpreted. The output from these two sector detectors is similar to the output of a resolver and is therefore often referred to as a resolver output. By referring to Fig. 18.59 it can plainly be seen that phase A leads phase B in time for clockwise rotation and that phase B leads phase A in time for counterclockwise rotation. A little study will show that this is true irrespective of the rate of change of θ.

Zero-crossing detectors and logical circuits can be used to give four outputs per cycle along with the sense of rotation. A schematic of a circuit which performs this function by means of derivative logic can be seen in Fig. 18.60. A set of clocked nonderivative logic which will perform this function is outlined below.

$$ {}_1a_1 = Ac \qquad {}_0a_1 = A'c, \; {}_1b_1 = Bc \qquad {}_0b_1 = B'c $$

and

$$ \text{1 unit clockwise} = AB'B_1c\; A'BB'_1c\; BAA'_1c\; B'A'A_1c $$

$$ \text{1 unit counterclockwise} = ABB'_1c\; A'B'B_1c\; BA'A_1c\; B'AA'_1c $$

These four points per cycle are not necessarily of equal weight, for the distance represented by the gaps between them is determined by the form, shape, and placement of the sector detectors and the sensitivity and backlash of the zero-crossing detectors. An unobvious feature of the zero-crossing detector which is imperative

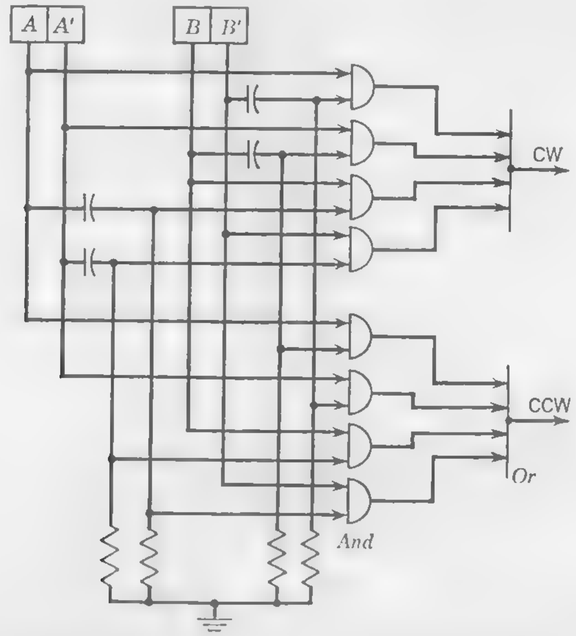


FIG. 18.60. Schematic using derivative logic of a circuit which yields four pulses per revolution and sense.

for nonambiguous operation of these incremental position-to-number converters is the backlash. Backlash is absolutely necessary, for without it the time constants of the logical circuits could be such as to produce errors if the shaft were to set at this zero value and jitter. The special features of this device are the simplicity of datum setting due to the ability to clear the counter at any position. The disadvantage is the possible datum shift due to pulse losses from such things as power interruptions.

Sweep-count Devices. These devices count the segments of a revolution from datum on a second constantly rotating shaft until its position is coincident with that of the input shaft. The counters utilized in these devices are unidirectional, redundant and the loss of pulses is of less importance for the value is ascertained with respect to datum once each sampling period. Devices of this nature have been made without step-up gearing to an accuracy of 1 part in 5,000 and sampling rates of 60 per second. Electric, magnetic, and photoelectric segments are those commonly counted.

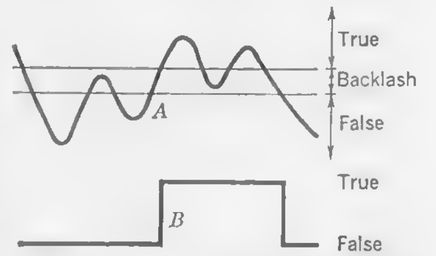


FIG. 18.61. Waveforms showing transfer function of the d-c trigger pair including backlash: A, input; B, output.

Resolver Phase-shift Converter. A simple electromagnetic resolver can transform an input voltage $A \sin wt$ and shaft angle θ into two output voltages $A_1 \sin \theta \sin wt$ and $A_2 \cos \theta \sin wt$, or it can transform two input voltages $B_1 \sin wt$ and $B_2 \cos wt$ and its shaft angle θ into an output voltage $B_3 \sin wt \sin \theta + B_4 \cos wt \cos \theta$; and if B_1 and B_2 are so adjusted that $B_3 = B_4$ then this output is equal to $B_3 \sin (wt + \theta)$. The accuracy of this transformation is a function of the input waveforms and of the resolver.

The amplitude ratio, the phase shift, and the harmonic content of the two input voltages must be very well controlled.

Simple Timing Determination. If a gate is enabled between a clock source and a counter when the reference voltage passes through zero (positive-going), and the gate is disabled when the resolved voltage passes through zero (positive-going), the total count racked up will be equal to θ divided by the total number of pulses per cycle. This total number of pulses can be made exactly equal to 2^n by counting the clock down by 2^n and forming the reference voltage from this period.

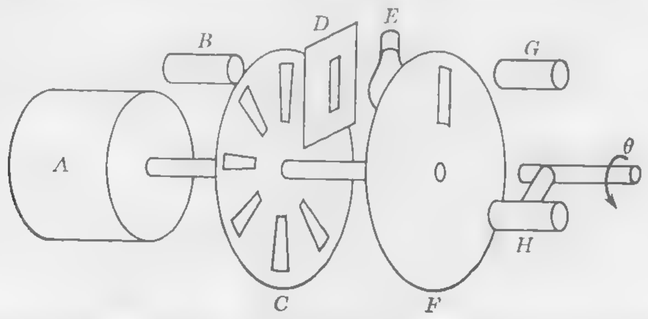


FIG. 18.62. Photoelectric-sweep count mechanism: *A*, motor; *B*, counting photoelectric cell; *C*, sector-counter disk; *D*, slit; *E*, light; *F*, reference slit; *G*, reference photoelectric cell; *H*, input photoelectric cell; θ, input shaft angle.

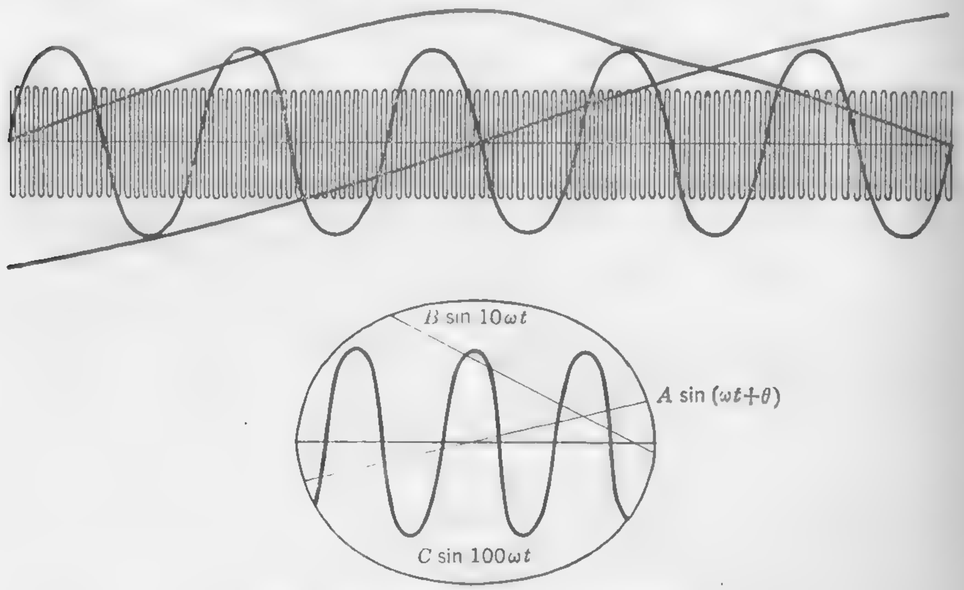


FIG. 18.63. Resolver vernier waveforms.

Vernier Techniques (Decimal). If the reference sinusoids $A \sin wt$ and $A \cos wt$ are plotted against one another on an oscilloscope a circle results. If the cathode beam is disabled except when the resolved phase-shifted voltage passes through zero (positive-going), a bright spot will appear on the oscilloscope screen at the angular position θ. Expressed in decimal form $\theta = a_n, a_{n-2}, \ldots, a_a, a_1$ where the a's can only have the decimal characters 0 through 9. If the reference voltages are each frequency-multiplied by 10, held at zero reference phase, plotted against one another, and the beam is pulsed on when phase-shifted nonfrequency multiplied voltage passes through zero (positive-going), a spot will appear on the scope at an angle of $\theta - a_n 10^n$. If the reference frequency used above is again multiplied by 10 and held at zero reference, and this and its cosine are cross-plotted by the deflection circuitry of still

another cathode-ray tube, and the beam pulsed on when the phase-shifted nonfrequency multiplied voltage passes through zero (positive-going), a spot will appear on the screen at an angle $\theta - a_n 10^n - a_{n-1}10^{n-1}$. This frequency multiplication cross plotting and strobing can be profitably continued until the full precision is used up. If collector plates were decimally placed in a circle around the face of the cathode-ray tubes the beam conductivity of each tube would denote the number corresponding to the a's representing the angle θ. The control would have to be extremely precise

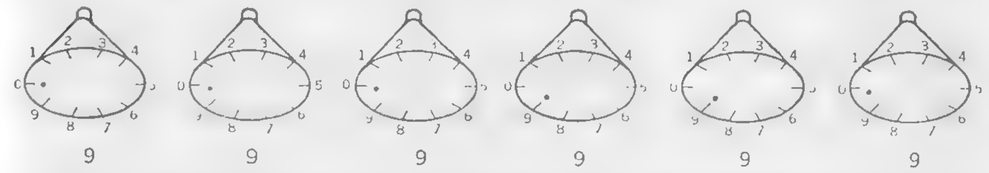


FIG. 18.64. Cathode-ray-tube faces including spots for a three-decimal digit display showing the strobed position both without and with subsequent digit-value biasing.

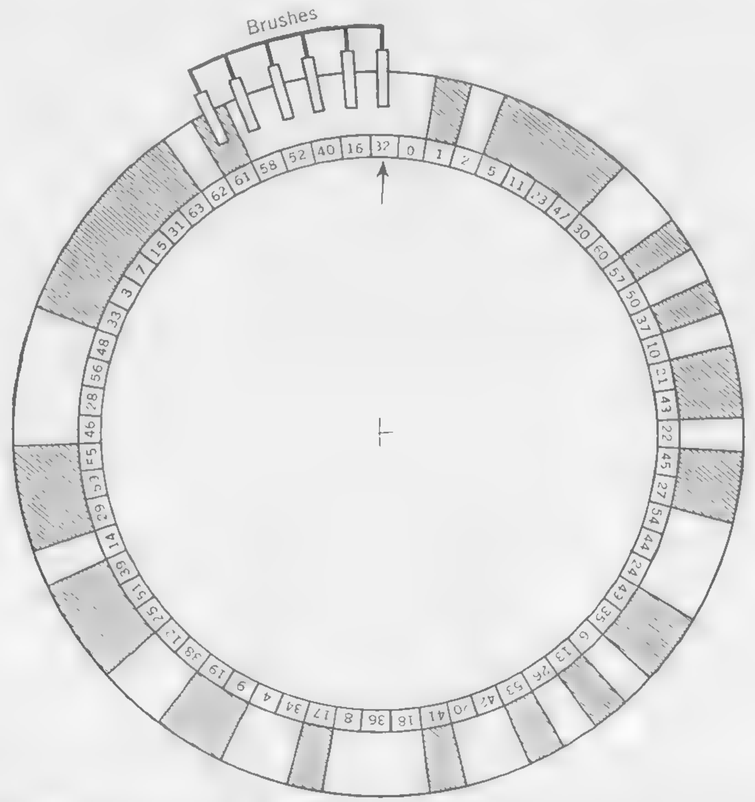


FIG. 18.65. Single-channel code wheel of 5 bits with parallel read-out. As the wheel rotates by the brushes the code contained on the brushes corresponds to a binary number equal to the decimal value appearing at the arrow.

in order to quantize these beam conductions. In order to alleviate this quantizing problem the value of a subsequent digit can be subtracted from the phase-shifted sinusoid before it enters the zero-crossing detector which forms its beam pulse. If this compensation is utilized with proper scaling the beam will fall at the same point on the target for all values of θ.

Resonant Members. The transverse resonant frequency of a rod, reed, or string is a function of the length of the string. If a member is caused to change in length

then the resonant frequency will change, and if the beam is excited into resonance and the number of traverse vibrations are counted then this count is a function of the time integral of the frequency.

Code Wheels. Code wheels have been made using all sorts of binary pickups, such as brush, switch, photocell, capacitive, or inductive pickups. Single binary wheels have been made which divide a single revolution into 2^{15} distinct sectors. Many revolutions can be encoded if more than one code wheel is used with gearing between them. A special code can be used which has properties such that only one channel needs to be read in order to determine uniquely the position of the shaft (see Fig. 18.65).

When the single-radix binary code is used there will always be ambiguous output at the carries, for the binary detectors will not switch instantaneously and in unison. Several methods of getting around this are possible. One of them involves a special coding and the other standard coding but special switching.

Carry-ambiguity Suppression. Probably the simplest carry-ambiguity suppression method to understand is the use of a second code wheel shifted one-half bit out of register with the first, thereby ensuring that when one wheel is propagating a carry the other is not. The selection of which code wheel to read can be done on the basis of the value of the least significant digit of either. The possible ambiguity at switch time in the least significant digit causes no trouble, for it is within the precision of the device no matter which value is taken. Care must be taken in order that back circuits do not interconnect the two wheels. This is commonly done with diodes or relays. The two disks are commonly one with two sets of brushes appropriately phase-shifted from one another. The position accuracy of the brushes can be lessened with no loss in over-all accuracy if the appropriate brush for the mth digit is selected by the value of the m-1th digit and the shift for the mth digit pickup is $\frac{1}{2}$ (m-1th unit). This is called "V-scan."

It should be noted that even though the pickup is binary the coding could have been any binary code utilizing an alternating least significant digit.

A method often used is to knife-edge toggle the detectors and to inhibit the encoder's output during the carry. Another method often used is the physical motion of all but the least significant bit pickup just before read time in accordance with the value of the least significant bit.

Cyclic Codes. These codes all have a form such that at most one bit is changing at a time. A simple cyclic binary code is shown in Fig. 18.66. A method of decoding

0	00000	16	11000
1	00001	17	11001
2	00011	18	11011
3	00010	19	11010
4	00110	20	11110
5	00111	21	11111
6	00101	22	11101
7	00100	23	11100
8	01100	24	10100
9	01101	25	10101
10	01111	26	10111
11	01110	27	10110
12	01010	28	10010
13	01011	29	10011
14	01001	30	10001
15	01000	31	10000

FIG. 18.66. A 5-bit reflected binary code.

this which utilizes the data in a serial manner most significant digit first is shown in Fig. 18.67. Think of the single-radix binary number as being *NRZ* codes, most significant digit first; then think of its derivitive as being a full-wave rectifier *RZ* coded number, most significant digit first. This number is a cyclic number. Most

conversions from this cyclic number system to normal binary function are done as the above example depicted.

18.6.3. Time and Frequency. *Time-to-number Converters.* A time interval can be easily converted to a number by counting the number of equally time-space-clock pulses which occur during the time interval. This number times the period of the clock pulses equals the time interval. In order to have high precision for short-duration time-interval measurement it is necessary to have a very-high-frequency clock source or to resort to vernier techniques.

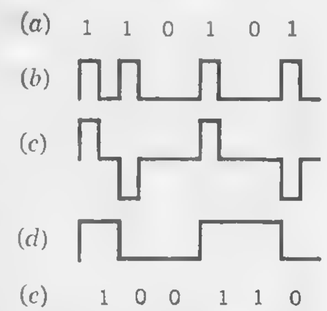


FIG. 18.67. Reflected-to-binary conversion waveforms for the number "38": (*a*) reflected binary number; (*b*) *RZ* coded reflected binary number; (*c*) *B* with every other pulse inverted; (*d*) *RZ* coded *C*, also *RZ* coded, single-radix binary number; (*e*) binary number.

The vernier pulse-timing method is generally one in which the position of a pulse between two clock pulses is determined by means of a secondary clock source whose frequency is quite close but not equal to the primary clock source. The methods of computing this use a vernier-pulse source in which m vernier-pulse periods are equal to $M + 1$ clock-pulse periods. The 11-10 verniers common in inch and metric measurement operate via the same principles. It can be seen from the two top waveforms of Fig. 18.68 that the errors in direct measurement are t_1 plus t_2, or t_1 plus t_3 minus one clock period, according to the method used.

As can be seen from the next three waveforms in Fig. 18.68 the coarse error t_3 can be determined by means of the vernier pulses which start at the terminating pulse. When a vernier pulse and a clock pulse coincide the number of vernier pulses is its fractional measure of $P_c - t_3$. This method initiates clock and vernier pulses at the instant of the initiating and terminating pulse, respectively.

Two possible errors exist in this system, and they are the pulse-train-starting error and a conceivable loss of one count for t_2 due to variations in switching time when the time being measured is nearly an integral number of clock periods. The latter can be completely overcome and the former reduced by using the method shown in the bottom three waveforms of Fig. 18.68. The pulse-train-starting-time error is reduced, for two identical ones are used, each of which cancels out the other. The method to be described is useful only for measuring times greater than M vernier pulses where $1/m$ is the vernier ratio.

A count of the vernier pulses from initiation to vernier-clock coincidence is a fractional measure of t_1 and a count of the vernier pulses from termination to vernier-clock coincidence is a fractional measure of $P_c - t_3$. In order to obtain an unambiguous clock-pulse count it is necessary that pulses not be counted in the vicinity of initiation and termination. This can be accomplished by counting vernier pulses until vernier-clock coincidence (number of vernier pulses ≡ proper number of clock pulses), then counting clock pulses to termination vernier-clock pulse coincidence, then counting vernier pulses until the Mth termination vernier pulse. This yields $M + 1$ more pulses than the number of clock pulses contained in t_2. This $M + 1$ excess is always present and can be readily subtracted, or the clock-pulse counter can have it ($M + 1$) for an initial negative bias.

Though vernier methods allow operation at a lower frequency there is nothing in them that allows more timing errors than direct-count techniques. In fact, since coincidence is detected instead of count-gated, and since accurate timing waveforms must be generated of arbitrary phase, the timing problems are amplified.

Frequency-to-number Converters. There are at present no direct instantaneous frequency-to-number converters available. The direct frequency-to-number converters which do exist count the number of pulses of the input frequency which occur in a given time interval. When this is properly scaled the count is the frequency. Devices can be built which measure the instantaneous period of the input frequency, and this is the reciprocal of the frequency. Frequency to shaft position or voltage

and then to number converters can be made. The integral of frequency with respect to time can be accumulated in a counter which counts cycles.

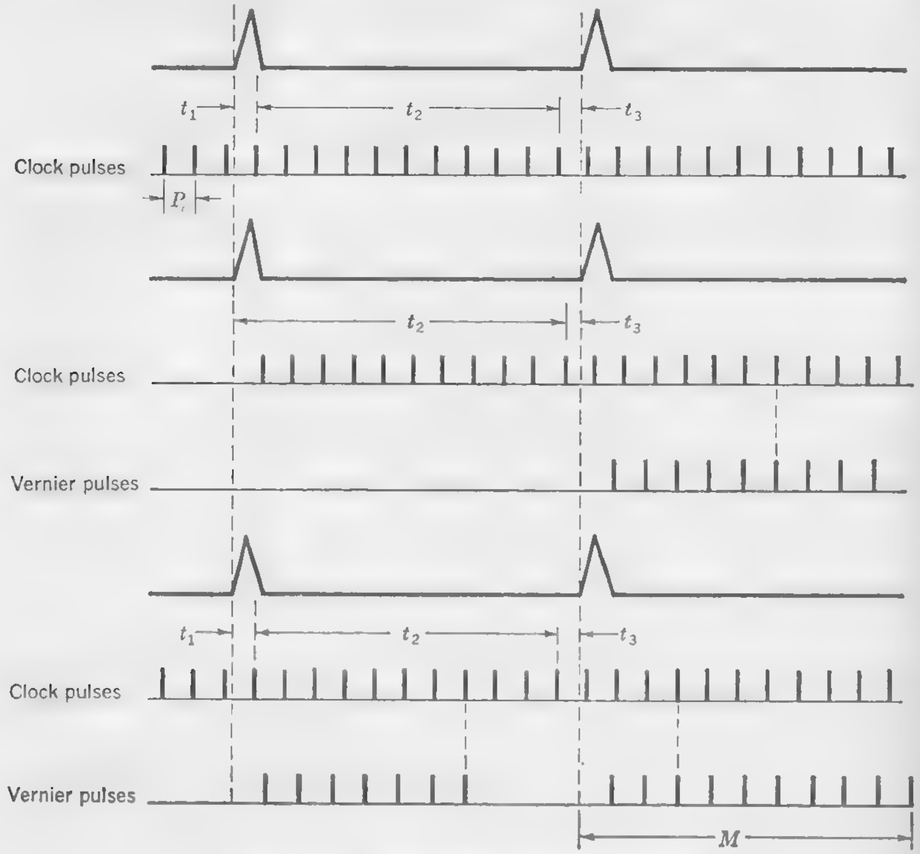


FIG. 18.68. Vernier pulse-timing waveforms.

18.7. DIGITAL-TO-ANALOG CONVERTERS*

The conversion from digital (quantized, coded, and sampled) data to analog form is one that is necessary for on-the-line computers, for plotters, and for many automation systems.

These devices perform the complementary function of the analog-to-digital converters (Sec. 18.6). Any operation that was performed in the forward loop in Sec. 18.6 can be placed in the feedback loop of a servo, and the device will then be capable of operating as a digital-to-analog computer. In general the analog output is desired as a continuous function, and therefore smoothing must be applied, for the digital data are always discrete. When a single converter is being multiplexed between many outputs the smoothing should not be performed inside the converter but rather after the separation switch. Some converters are inherently of high-frequency response, and others inherently contain considerable smoothing. All these converters are essentially weighing or counting devices. The precision, stability, accuracy, sampling rate, and response time are all extremely important properties of each.

The problem of converting from machine language to converter code is often a difficult one, but for true binary machines the problem does not usually exist, for most decoders can operate very well from the binary code. A few of the decoders are quite well suited to other codes, and some of the decoders operate incrementally so that any counting code offers no difficulty. Very few of these decoders are of

* By William A. Farrand.

the sweep-balance type, and therefore very few have sufficient frequency response to jump indiscriminately about in their range on successive samples.

18.7.1. Number-to-voltage Converters. Voltage has been used as an intermediary in the measurement of almost all variables in nature. The high-frequency response and ease of construction of circuits which transform voltage are well known. Many of the converters derive a voltage at an instant, and if this voltage is not clamped and held until the voltage is derived from the next sample, the output will be useless. Clamps are discussed in Art. 18.6.1.

Serial Binary Weighted Discharge. The device makes use of a condenser and decay circuit such that its voltage will be reduced to one-half during each clock period. The circuit is charged with a given number of coulombs each clock time, as the serial number is read least significant digit first. It is charged only at a specific clock time if the value of that particular digit is 1, and is left alone if the particular digit is 0. Since the nth significant digit has had its charge cut in half $n - 1$ times when the most significant digit charges the condenser, the value of voltage on the condenser will be proportional to the serial binary number which controlled its charge. Another binary weighted-discharge technique, which is not so dependent on absolute time and on decay constants, consists of one condenser which is charged or discharged dependent upon the value of the binary bit each clock time. Between clock times it shares its charge with

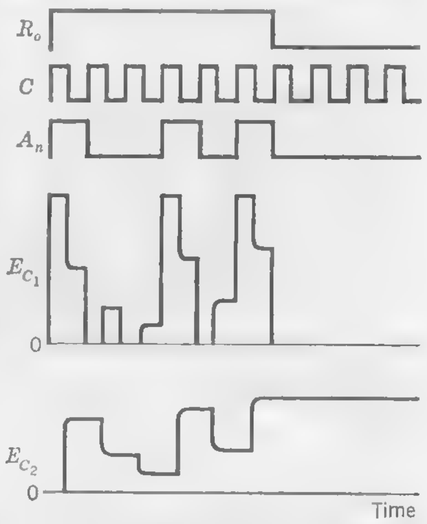


FIG. 18.70. Waveforms in the serial binary weighted-discharge circuit.

FIG. 18.69. Serial binary weighted-discharge circuit.

that of another condenser of equal capacitance. Its theory is similar to the RC case above, but the discharge is of staircase instead of exponential form and voltage, not coulomb charging, is required.

It is conventional to clamp this voltage one clock time after the most significant digit has charged it and to maintain this voltage in the clamp until the next number is decoded. In this manner a voltage can be generated such that at the end of each number interval it is of value corresponding to the serial binary number. The staircase circuit has a built-in output clamp, the capacitor.

Digital Potentiometers. A digital potentiometer can be defined as a number-to-voltage ratio converter which uses resistors as the voltage control. The most accurate of these devices convert from a parallel contact-closure code through a weighted-resistor network. These contact-closure devices though of excellent accuracy are definitely restricted in speed of operation. When high operating speed is desired electronic switches are utilized in place of the contacts. The accuracy of the weighting is not so severe as might be expected for the least significant bit weight could be in error by 100 per cent and contribute only one least-count total error.

There are three general methods of resistor decoding, voltage summation, current summation, and simulated potentiometer. In a voltage-summation circuit the load remains fixed, for the resistors tie either to datum or to reference. In a current-summation circuit the summing point is forced to zero by an amplifier and a feedback resistor. In a simulated potentiometer the resistors are assembled into a ladder in such a manner that the impedence ratio depicts the number being decoded. When extreme accuracy is desired the resistor decoding net should be used in a current-

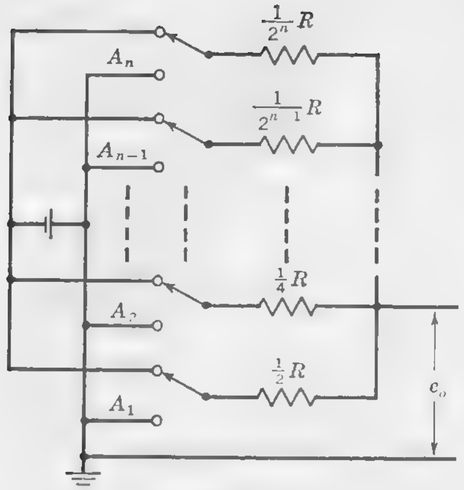


FIG. 18.71. Voltage-summation network $e_0 = A_n 2^{n-1} + A_{n-1}2^{n-2} + \cdots + A_2 2^1 + A_1$

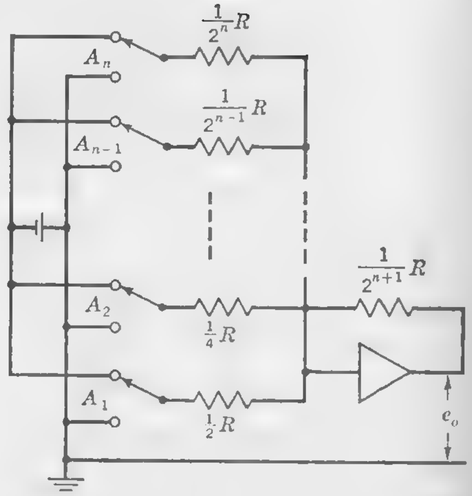


FIG. 18.72. Current-summation network $e_0 = A_n 2^{n-1} + A_{n-1}2^{n-2} + \cdots + A_2 2^1 + A_1$

summing circuit followed by a drift-stabilizer operational amplifier, and great care should be taken with respect to guard ringing.

Sequential Control. A register which is logically interconnected to operate as a counter can be used as a digital potentiometer if a resistor matrix is appropriately attached through suitable switches. This type of potentiometer can be set to give any output in its range by cycling it to the prescribed value.

A binary-coded commutator with resistor decoding constitutes an analog-to-digital-to-analog converter with the digital data sequentially controlled.

Coded Control. A register which can have its members individually controlled can be used as a digital potentiometer if a resistor matrix is appropriately attached through suitable switches. A binary-coded decimal resistor decoder of three decimal digits is shown in Fig. 18.75. A reflected binary-to-analog-voltage converter using double-pole double-throw relays as the register is shown in Fig. 18.76.

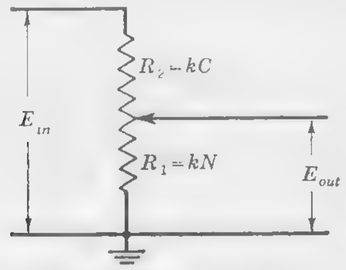


FIG. 18.73. General potentiometer form. N is the number. C is the complement.

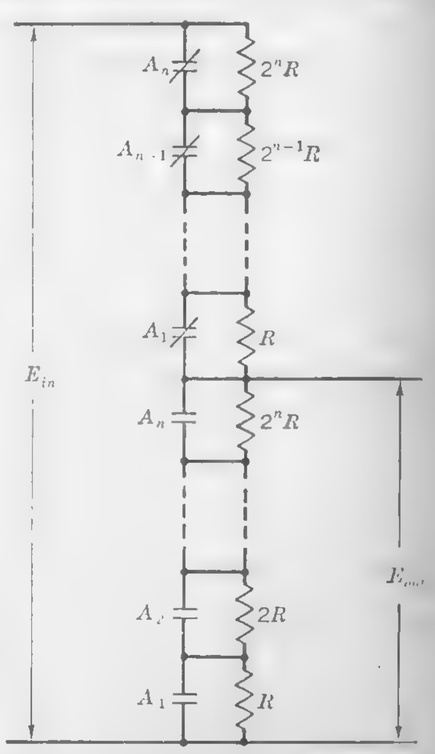


FIG. 18.74. Single-radix binary-number decoding potentiometer circuit.

Number-to-boxcar-to-voltage Converters. This type of converter forms a rectangular pulse of fixed amplitude and of variable length (controlled by the number being converted) at a fixed repetition rate into a low-pass filter whose output is then proportional to the number being converted. The most convenient method of forming the boxcar is to form two pulses separated by the number of clock pulses indicated by the number to be converted. These two pulses are then used, by means of an electronic switch, to gate the fixed voltage into the filter. The repetition period of this process must not be in excess of the number of clock pulses representative of the largest number to be converted. The filter must be extremely well designed or the output ripple will be too great (more than one bit).

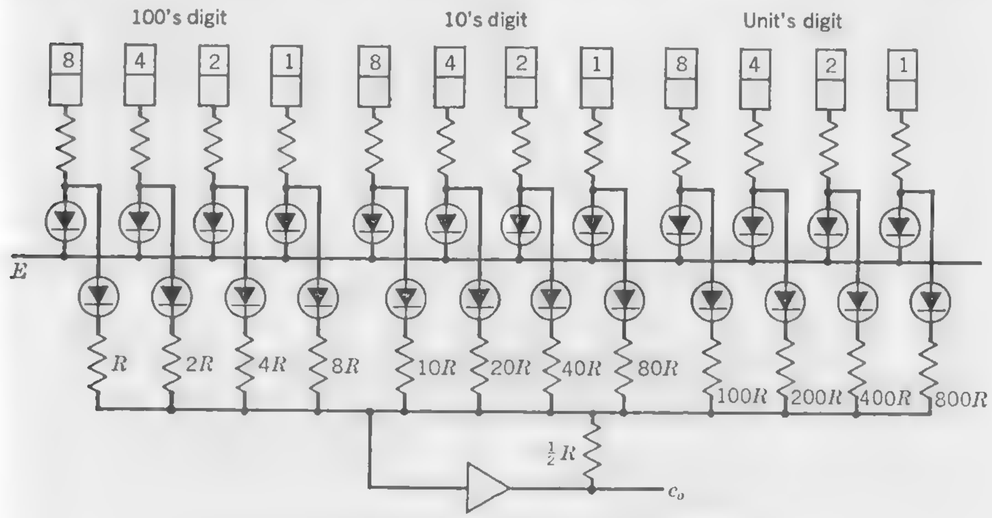


FIG. 18.75. Binary-coded decimal resistor decoder.

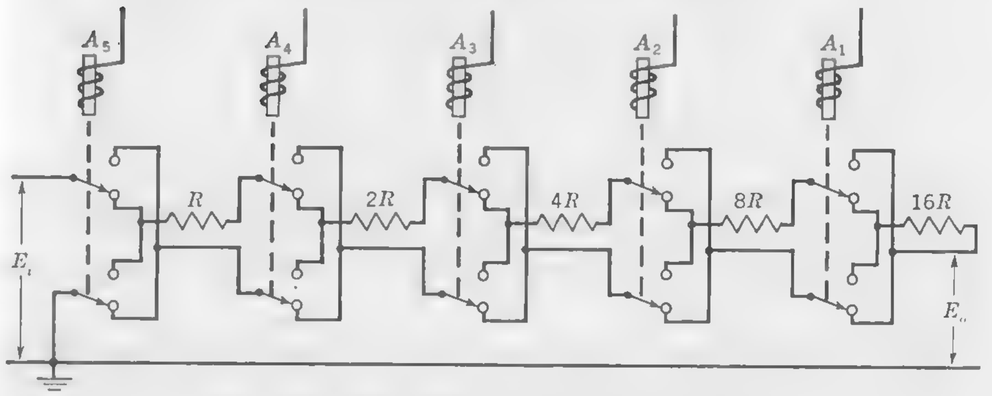


FIG. 18.76. Relay reflected binary voltage decoder.

If a magnetic drum is used in the converter one drum revolution can represent the period of one cycle of the boxcar, with the origin pulse enabling the reference-voltage gate, and number coincidence with clock-pulse counter or number coincidence with a drum register disabling the reference-voltage gate. In this type of device the speed of the drum does not have to be precise on a long-term basis; it only has to be uniform over a single drum revolution.

Number-to-position-to-voltage Converter. Code-wheel servos, step-motor servos, or open-cycle step motors followed by potentiometers form number-to-position-to-voltage converters. The code-wheel servos, the step motor, and step-motor servos are described in Art. 18.7.2. A code-wheel servo can utilize the code wheel as a feedback element for the servo and also as the register and switch for a resistor-decoding matrix.

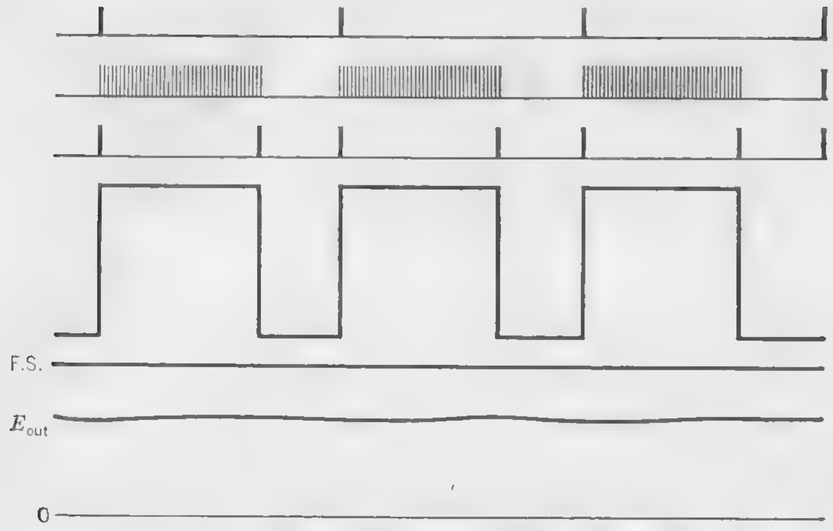


FIG. 18.77. Number-to-boxcar-to-voltage converter.

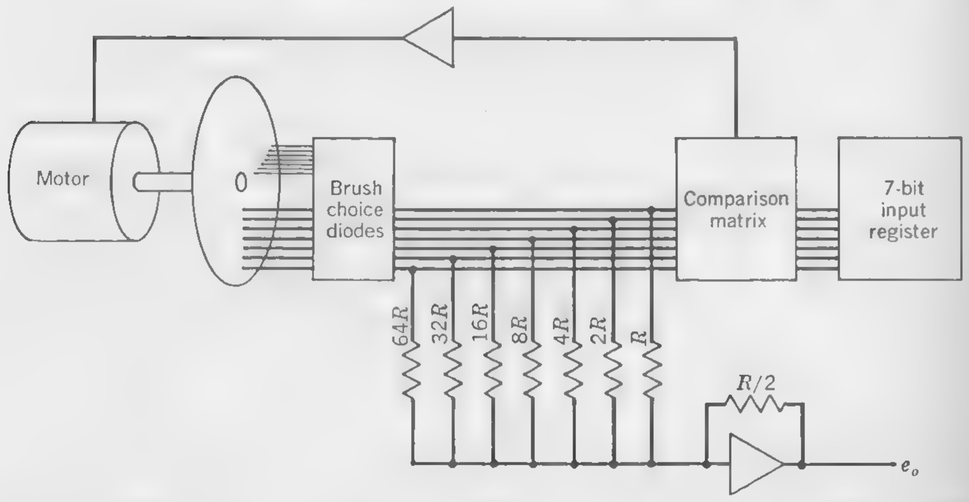


FIG. 18.78. Code-wheel servo using code wheel as the register controlling resistor decoding matrix.

18.7.2. Number-to-position Converters. For the convenience of discussion, position will be considered to be angular position, and for the instances where this is not correct the converters described can also be constructed as layout developments (planar representations) or the position can be transformed by means of various mechanical linkages.

Number-to-shaft position converters make possible the use of many conventional plotters for the graphical display of computation results. In simulators, process-control applications, and other "on-the-line" installations the positioning of a shaft is one of the most important functions of the computer. The frequency response is generally quite low but the accuracy can be exceedingly high, and if the device is used to follow a function the low-frequency response of the converter is not always the determining factor in system frequency response.

Incremental Converters (Step Motors). Solenoid-ratchet Motors. Most step motors derive their torque from solenoids which are pulsed. In some cases the solenoid is directly connected to the ratchet; in other cases the solenoid pulls the pawl against a spring and the spring then forces the ratchet wheel around on the return stroke.

The relative merit of these two mechanisms is a function of their over-all design, for with a spring-type detent the second mechanism gives an output torque which is more nearly constant than the first though utilizing a slightly more complicated mechanism. The large amounts of energy necessary for solenoid actuation make it difficult to control these by means of normal low-powered computing elements. Relays, thyratrons, and magamps are usually used.

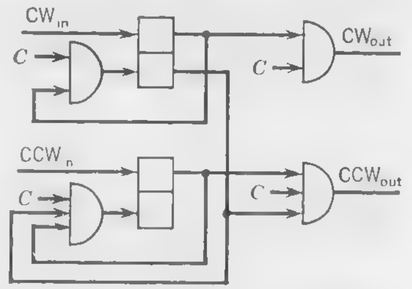


FIG. 18.79. Input asynchronizer.

Bidirectional step motors can be constructed from simple (unidirectional) step motors by intercoupling them through a differential or by having their separate solenoids each capable of turning the same shaft but in opposite directions. With the latter type of bidirectional motor, care must be taken to asynchronize the two inputs, for if either solenoid is activated while the other is operating erroneous results may occur. In fact, in some designs the device will jam.

Spring type detents are commonly used to insure that the output shaft is driven only by the solenoid. It is common for the initial motion of the solenoid to apply an additional detent which only its final motion can overcome.

Energy-release Motor. These devices normally consist of a prime mover and an escapement. They are capable of very high torques and very high operating speeds with very little complexity. The disadvantages are shorter life due to a constantly running motor and low efficiency in low-duty-cycle operation. A common high-acceleration energy-release control consists of a simple gear differential and two brakes, one on each side wheel. The prime mover drives the spider and the side wheel attached to the load has its brake full on in the deenergized position. When the prime mover is to be connected to the output, the other side wheel of the differential is braked to a stop and the load side brake is released. If this braking and unbraking are controlled to give a fixed quantum of output motion an excellent step

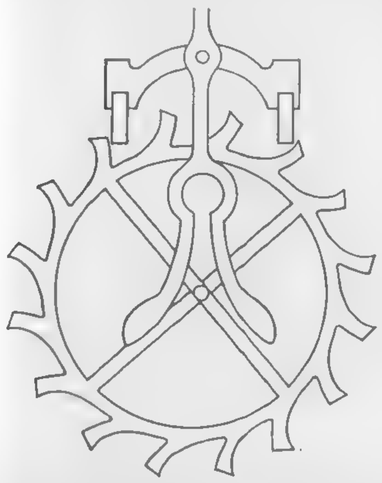


FIG. 18.80. Watch escapement.

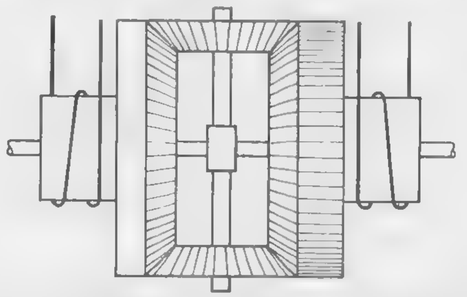


FIG. 18.81. Differential with brakes on both sidewheels.

motor results. For bidirectional operation two of these differential escapements are used on one prime mover. The characteristics of this device can be such that arbitrary input to the clockwise and counterclockwise controls causes no difficulty in the output. The quantum of displacement release can be set by the brake if it is a dog clutch.

A-C Synchronous Motors. A permanent magnet or constantly excited wound-rotor two-phase electric motor can operate as a stepping motor if the excitation to the two fields is properly sequence-controlled. The proper control and a relay sequencer are shown in Fig. 18.82. This device uses magnetic detents for its operation.

A single-phase A-C synchronous motor with notched permanent-magnetic rotor will also work if permanent-magnetic field pieces are inserted in the stator in such

a manner as to lead the wound-stator poles. These permanent magnetic field pieces ensure the direction of rotation of the motor and provide the detenting force for its shaft.

Number-to-voltage-to-position Converters. These devices conventionally take the form of a voltage-to-position servo preceded by one of the number-to-voltage converters discussed in Art. 18.7.1. Many positioning problems have torque, frequency response, space, and environmental requirements such that a voltage-to-position servo can be used with a minimum of development. For these and for situations in

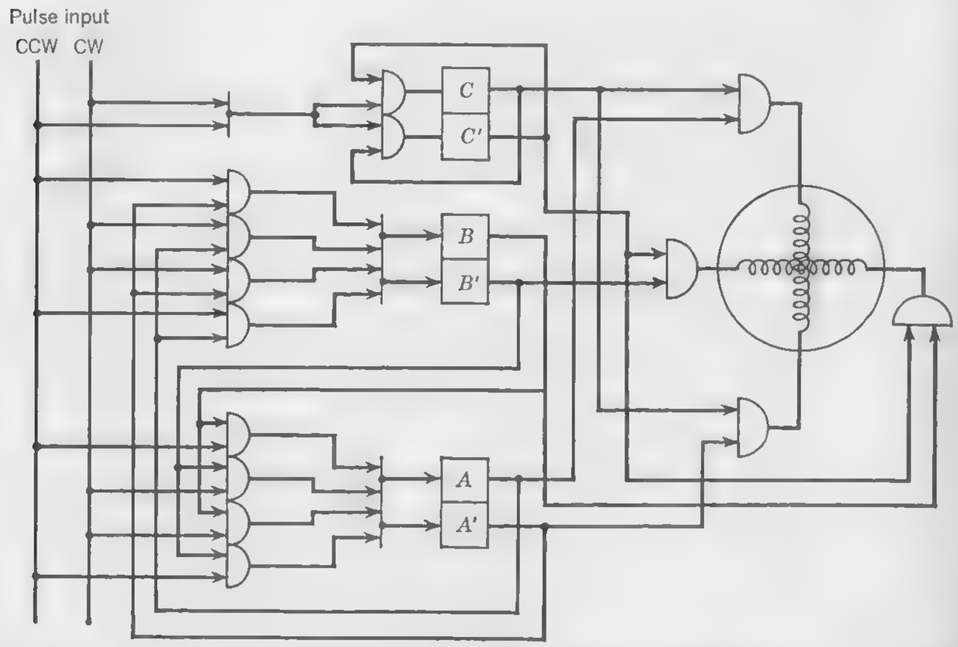


FIG. 18.82. Two-phase d-c synchronous motor bidirectional impulse sequencer: A, phase A control flip-flop; B, phase B control flip-flop; C, direction-control flip-flop.

which such a servo already exists on the shaft a number-to-voltage converter can be used to develop the servo's input with good over-all performance. Accuracies of the order of 0.05 per cent have been obtained by this method on single-speed systems. When multispeed servo systems are utilized special coding between the number-to-voltage converters can increase the accuracy. The voltages generated from the number may take many forms and for low-accuracy work resolvers and synchros are sometimes simulated.

Code-wheel Servos. These devices utilize the code wheel-to-number converters described in Art. 18.6.2 and electromechanical servos which position the code wheel to the appropriate number. One interesting characteristic of these devices is the possible elimination of the carry-ambiguity-suppression equipment by means of appropriate servo design. If the servo can be so designed as to overshoot any carry ambiguity and yet settle on a single value even when but one count away from a carry then no error will exist in the output of this converter in spite of the ambiguous carries which the code wheel emits.

Step-motor Servos. If a bidirectional step motor is capable of emitting a pulse when its shaft is moved one step clockwise and a separately distinguishable pulse when it is moved one step counter clockwise, then these pulses can be accumulated by adding or subtracting into a register, and power pulses fed to the step motor in such a manner as either to null the register or to make its value equal to that of the input number. This entire mechanism is then a true step-motor servo with good reliability, even though the step motor may fail to actuate on a few of its inputs.

18.7.3. Time and Frequency. *Number-to-time Converters.* A register containing the number to be transformed to time can be counted down by equally time-spaced clock pulses, and when the counter is at the value zero the time duration is accurately the number of pulses times the time duration between pulses. The accuracy of this is limited only by the speed at which the register may be counted down and the time duration between the pulses. The spacing between the clock pulses can be controlled very accurately (1 part in 10^9 is now being controlled in the field).

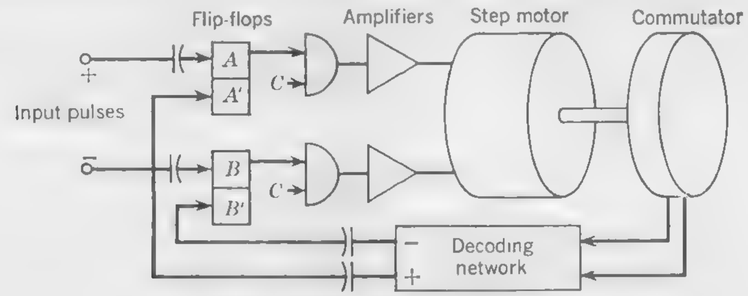


FIG. 18.83. Step-motor servo.

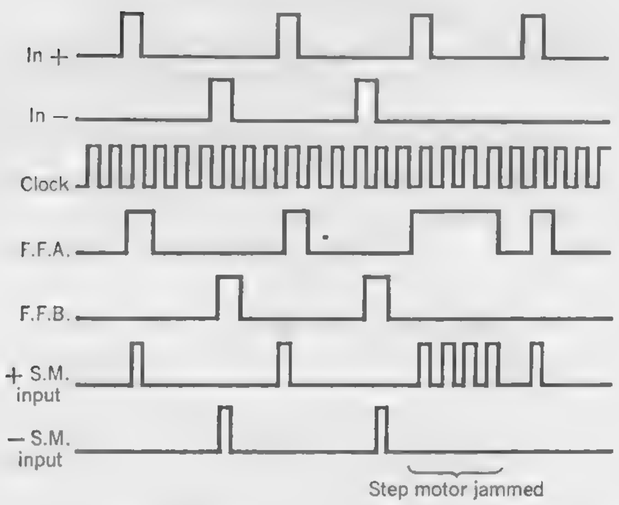


FIG. 18.84. Step-motor servo waveform with unreliable motor.

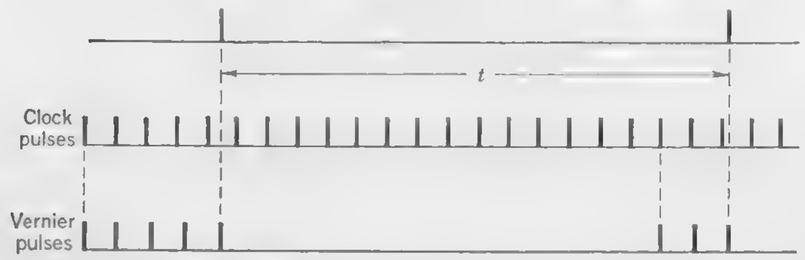


FIG. 18.85. Vernier number-to-time decoder waveforms.

Vernier techniques are useful when the necessary count rate is excessive because of either the accuracy or the brevity of the time interval. All the characteristics and methods discussed in Sec. 18.6 apply here equally well.

With magnetic-drum machines and other recirculating register machines the time from origin pulse to register value can be equivalent to the register's value. This is not normally an extremely accurate or necessarily an equipmentwise efficient operation unless the recirculating registers are very simple and their timing very precise.

Number-to-frequency Converters. Direct devices of this nature do not exist but are often called for. A frequent use is as an aid in first-order interpolation. The devices conventionally take the form of number-to-voltage or number-to-shaft position converters followed by voltage or shaft position-to-frequency converters. Voltage-to-frequency converters which utilize Miller feedback are commercially available at 0.05 per cent accuracy. Another indirect method is to convert the number to its reciprocal and convert this to time, immediately repeating the process so that this time is the period of the pulse train or square wave. When the frequency is high with respect to the time rate of change a servo to produce it can be formed. This may consist of an oscillator and a reactance tube whose output cycles are counted for a fixed interval and the control voltage to the reactance tube shifted in a direction to shift the frequency appropriately.

18.8. PLOTTERS*

A valuable aid in the solution of certain problems is often a cross plot of two of the problem variables. This plot, though seldom of full machine precision, often portrays more information than a simple tabulation of values, for it gives at a glance not only the value but also the derivative and past history. These data in plotted form readily point out relative maxima, relative minima, inflection points, limit cycling, oscillation, convergence, and divergence when present.

In the analog field many forms of plotters are readily available, and therefore the usual practice in digital work is to utilize a digital-to-analog converter and one of the analog plotters. However, there are a few strictly digital plotters available and in some instances the analog field has made use of these digital techniques.

Most plotters utilize their base frame as reference and take measurements by one means or another along it. A few devices have been built which count the squares on the graph paper; this automatically corrects for paper warp or uneven ruling. A property of these devices which is sometimes valuable is their ability to turn out a functional plot by the simple use of appropriately ruled paper (e.g., a log plot on log paper).

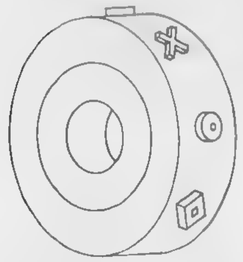


FIG. 18.86. An eight-symbol stamp record head.

18.8.1. Writing Means. The digital data being recorded are the coordinates of a series of discrete points. The plotter may, however, plot the discrete points or it may trace a curve through them. The form of the curve is then a function of the response of the recorder and the time and position displacement of the points. If it plots the points discretely and the symbols plotted are selectively distinguishable then more than one curve may be plotted in a single set of coordinates and it is even permissible to intermix the points being plotted from curve to curve.

The mechanism of writing determines to some extent many of the characteristics of the plotter such as frequency response, legibility of the plot, distinguishability of the plots one from another, viewability while plotting, permanence of the record, and ease of plotter maintenance.

The record heads can be honed to proper contact-face contour on most plotters by replacing the chart paper with crocus cloth and operating in some manner without "ink."

Symbol Stamps. The writing head of many plotters consists of a stamping device with means for selecting the appropriate symbol to be marked for the plot of that particular point. The selection scheme for the particular symbol varies greatly, but since the symbol choice is limited, simple solenoid, or solenoids and cams, normally suffice. The symbol selection is conventionally a fixed time sequence or a "select-upon-demand" selection.

If the slope of the curve can be plotted along with the coordinates of the point far fewer points than might be considered necessary for the conveying of the information

* By William A. Farrand.

may be plotted. A symbol stamp (θ) does this well when its center represents the point and its orientation represents the slope.

Ink Pens. Capillary flow is a common means of ink feed. Capillary-flow pens normally have a high mass and fair friction, for the pen is in forced contact with the paper. This limits the upper frequency response considerably. However, because of the mechanical gain available, ink-throwing pens are sometimes used. They have a special advantage in digital work, for the ink flow can easily be made intermittent.

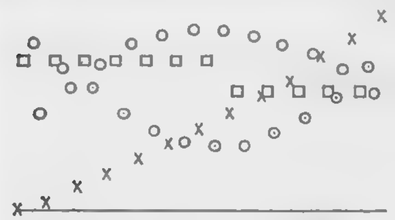


FIG. 18.87. A four-symbol plot.

FIG. 18.88. A plot of value and slope.

Writing with solids rather than liquids is quite common, the lead pencil, ball-point pen, and ball-point pencil being the most obvious of these methods. Normal graphite pencils have the disadvantage of wearing down and therefore are continually in need of adjustment. Ball-point pens have to date been rather unreliable slow-writing instruments. Solid leads, inks, dyes, and pigments have the great advantage of simple long-lasting "ink" supply.

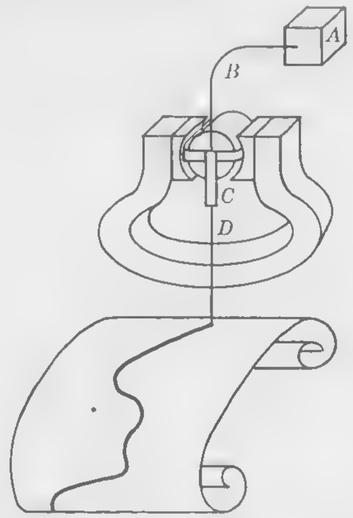


FIG. 18.89. An ink-throwing pen plotter: *A*, ink pump; *B*, ink feed line; *C*, ink nozzle mounted on galvanometer; *D*, ink.

Carbon Papers or Transfer Ribbons. This means of imprinting normally demands very large impression forces and therefore a low over-all speed. When carbon is used the production of multiple copies is simple, for an interleaved stack of carbon and paper may be formed with the thickness limited only by the impression force of the record head. Multiple copies are available when transfer ribbons are used, for an offset-printing master can be imprinted in this manner. The speed of imprinting a single copy through carbon transfer can be increased by raising the record-head temperature (for some carbons a 10°C rise decreases the legible imprint force by a factor of 100). Carbon paper is paper-coated on one side with a mixture of waxes and oils into which color is added, not necessarily carbon. Inks, dyes, pigments, and graphite are the usual coloring agents.

Electrosensitive Paper. These papers are marked permanently and indelibly by passing current through their thickness. The intensity of the marking is dependent on the

writing current, making various contrasts, even halftones, available. The required voltage as well as the current for marking are a function of the writing speed. Figure 18.90 shows a representative curve of impedance and current vs. writing speed for a small-area writing pen. When symbol stamping this paper it is normally of value to dissect the stamp into several individual circuits so that a point breakdown of the paper will not reduce the voltage on the rest of the stamp to a value below the breakdown threshold. These papers are normally insensitive to light and relatively unaffected by heat.

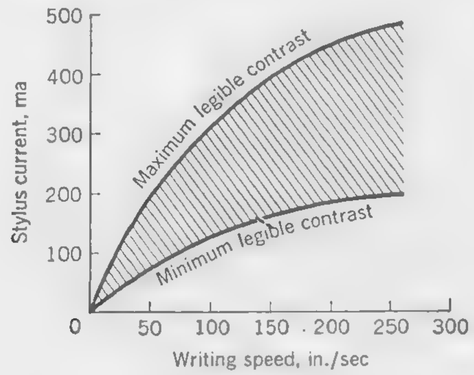


FIG. 18.90.

Dry Electrosensitive Paper. A typical dry electrosensitive paper consists of an aluminum backing on a conductive carbon-bearing paper with an opaque insulating electrosensitive coating on the face. This electrosensitive chemical breaks down when sufficient voltage is applied across it, making visible the black carbon underlayer. This is normally a nonelectrolytic process. In some papers the breakdown of this opaque insulating layer has as a by-product a poisonous gas. This gas is normally very low in concentration but in poorly ventilated cabinets could present a hazard. These papers stand up very well throughout the extreme ranges of temperature, and some are not affected by moisture. Papers are now available which can be used as masters for many of the offset-printing machines.

Moist Electrosensitive Paper. Many of these papers are moistened and their sensitivity is a function of their moisture content, for they contain electrolytic material. Since the papers are nearly dry in less than 10 min of normal air exposure the containers which feed the paper in strip-chart recorders or the total recorder in other recorders should be kept vaportight in order to ensure maximum sensitivity. The moist papers warp slightly upon drying, and in some instances it may be worthwhile to install calendaring rolls and heaters to alleviate this paper-warping tendency. If the moist electrosensitive paper has been exposed to the air for but a short time previous to marking, the environmental conditions of humidity and temperature throughout extreme ranges will not affect the legibility of copy.

Xerography. The literal translation is "dry writing," the lithographic nomenclature "dry writing by means of electrostatic charge." The basic process of xerography involves the electrostatic charging of a piece of paper in a selective manner and the brushing of this surface with oppositely charged powder (the dry ink). This powder adheres only to the charged areas and can be fused to the paper with heat.

Two techniques are usually added to the basic process, making a dry photographic reproduction using electrostatic techniques. The first of these is the use of a photoemissive material so that the entire insulator can first be positively charged and then selectively discharged by exposure to light. The second of these is the use of a positively charged copy sheet upon which the dry ink is transferred from the selectively charged plate and then affixed.

In plotters the write head will discharge the plate either electrostatically or by means of a light beam.

Ferrography. The basic process of ferrography involves the magnetic induction of a special sheet of paper in a selective manner and the brushing of this surface with a ferrous powder (the dry ink). This powder adheres only to the magnetized areas and can be fused to the paper with heat. In printing with this process it is possible to use unprocessed paper by selectively magnetizing a special drum and brushing its surface with the ink and then offset or impression transferring this to paper.

Photography. The literal translation is "light drawing." A sensitized paper normally coated with silver chloride or silver bromide when developed will produce a black image of the areas exposed to light and a white image of the areas unexposed. By a variation in the amount of light exposure a complete gradation throughout the gray scale can be obtained. Color processes are, of course, available by which the three primary colors or any combination thereof may be reproduced along with full intensity control. The grain size of the photographic process makes it possible to record large amounts of information in a very small space. The extreme light weight of the writing head makes possible excellent frequency response. The main disadvantages are the processing-time delays and the inability to view the plot as it is progressing.

Wax Papers. Wax papers, as discussed here, are not wax-impregnated. They are usually a dyed paper coated on one surface with a very uniform and thin coat of wax. These papers can be permanently and indelibly marked by rubbing their surface. Many of these require but ¼ oz of force to produce a thoroughly legible line with a pressure stylus, and even this force can be reduced by heated-pressure-stylus recording. A method of marking these wax papers without direct contact, and therefore very low drag friction, consists of an arc discharge through the paper, melting the wax and thus producing the mark. These papers are quite hardy and can stand humidity and temperature very well, as long as their melting point is not exceeded. The wax on most of the commercially available papers melts above 55°C (well above any normal room temperature). The very low temperatures do decrease the marking sensitivity, but this is seldom a problem.

Soot recordings have also been attempted, with some success. Their main attribute is the extremely negligible force which produces an excellent mark in pressure-stylus recording. These records can be made permanent by properly lacquering after imprinting.

18.8.2. Displacement Control. The problem of controlling the motion of the recording head with respect to the paper is very important, for it normally determines the maximum frequency response of the plotter. These controls are commonly two-axis record-head motion or single-axis record-head motion and single-axis paper motion. Two-axis paper motion is seldom used in digital plotting. It is not uncommon for at least one of the axes of the record-head motion to be only an apparent one and not an actual physical motion of the record head.

The choice of displacement control and mechanical configuration is as much a function of the personnel who operate the plotter as it is a function of the data to be plotted. Often such things as flat undistorted plots which can be viewed while being traced on peculiar-scale-factor preprinted paper are requested. This along with response time, data interplexing, curve definition, and over-all accuracy makes the plotter designer's task a difficult one.

Fixed Styli. In plotters utilizing fixed styli there are normally a multitude of styli positions across the full range of value of the variable, and the one corresponding to the variables value can be selected and caused to write. These styli normally use a purely electrical writing technique and therefore the common electrical selection schemes are used. When a mechanical motion of the selected stylus is necessary to cause writing, combs are commonly used as the selection mechanism.

Galvanometer. One of the most used displacement controls in recorders is a device based upon the principles of a moving-coil galvanometer. Usually, however, it is an extremely powerful movement and is commonly a servoed device. Since the galvanometer-type drive does not operate with an infinite lever arm there is some distortion. This is a sinusoidal compression of its variable and also a cosine cross coupling in the case of an orthogonal plot. Occasionally the galvanometer-to-pen

linkage is one which removes either the compression or the cross-coupling distortion or both.

Drums or Disks. This form of paper drive is very common. In fact the most plentiful plotters in production are the strip-chart or disk-chart recorders, and these utilize drums or disks as the basic paper drive. Since the plotter is usually located

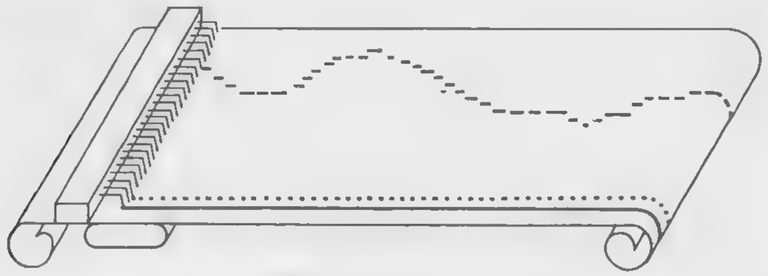


FIG. 18.91. A fixed-styli plotter.

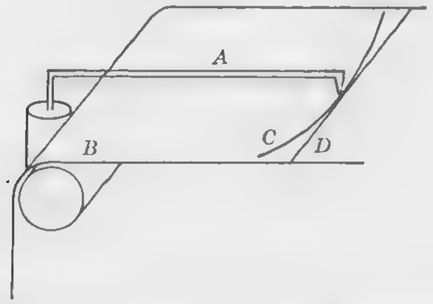


FIG. 18.92. Galvanometer plotter showing distortion: *A*, pen arm; *B*, paper; *C*, actual trace; *D*, orthogonal trace.

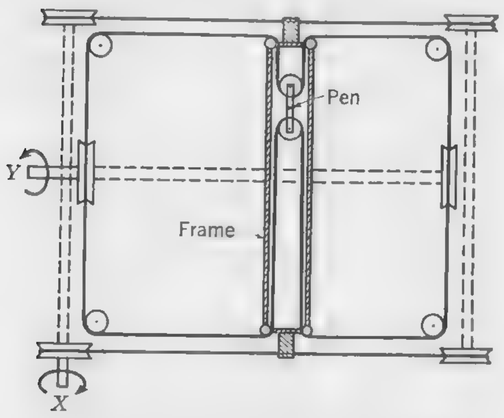


FIG. 18.93. A cable-driven *XY* plotter.

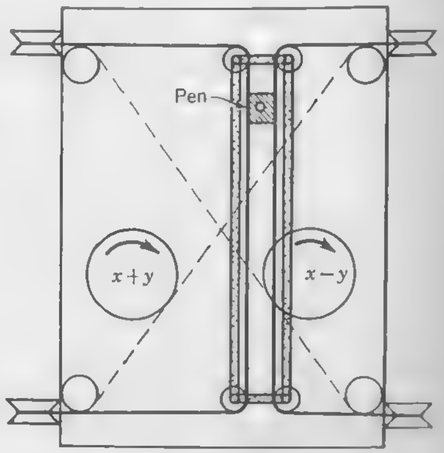


FIG. 18.94.

remote to the computer the positioning systems used include such devices as synchros, magnesyns, telegons, and resolvers, or coded combs or cams, step motors, or digital servos. Oftentimes gears, bar linkages, and string linkages are used between the prime mover and the drum.

Another drum-type drive places the paper on the inside surface of the drum and revolves the write head past it.

Cable Drives. Linear. When the record head can be constrained to have but one degree of freedom and this a simple translation, then the cable or band is an extremely practical drive linkage. This is true because of the extremely low mass of not only the drive but also the constraints. The cables are further extremely simple and can be utilized with very little backlash and in such a manner as to be extremely rugged.

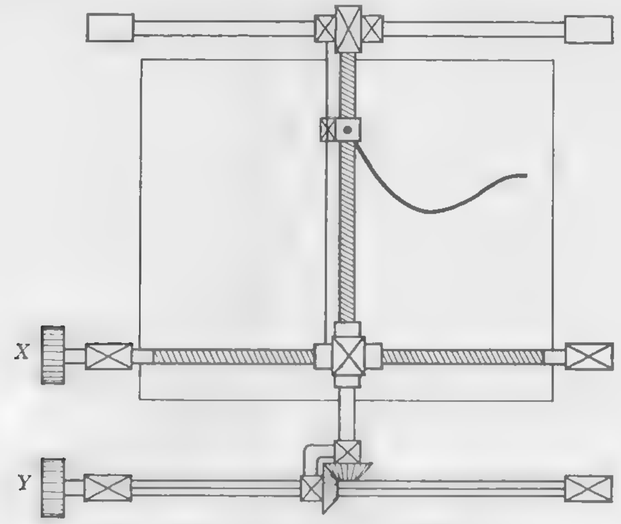


FIG. 18.95. Lead-screw and guide plotter.

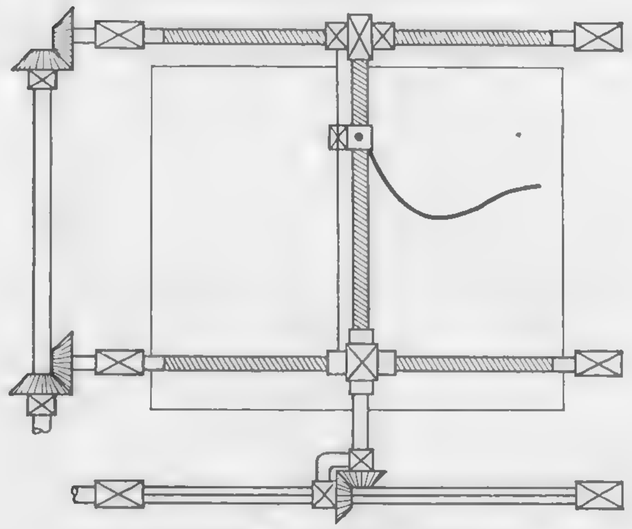


FIG. 18.96. Twin-lead-screw plotter.

Functional. Unique properties of string linkages have occasionally been utilized in the solution of the plotter design. One of the nicest of these systems for an orthogonal cartesian-coordinate flat plot is a simple mechanical drive system that can be obtained using an extremely light carriage. If the cable configuration is that shown in Fig. 18.94 and the drives are $x - y$ and $x + y$ instead of x and y alone, the record head is appropriately positioned.

Lead-screw Drives. A very direct and simple method of translationally positioning an object such as a record head is to place it on a threaded nut and to pass a lead screw through the nut. As the screw is rotated the nut is translated if it is constrained to not rotate.

Lead Screw and Guide. If a guide is used to mount the nut irrotationally a simple mechanism results which works very well as long as the distance from guide to lead screw is shorter than the guide-bearing length. If this is not true the system may well have positive feedback with respect to forces injected through coulomb friction in the guide. (This means that when the carriage starts to jam from static friction any force entered through the lead screw will only amplify the static friction.)

Twin Lead Screws. A second nut on a second synchronized lead screw may be the irrotational mount and with this the positive feedback is eliminated, but the problem of lead-screw whip and alignment is emphasized. However, any backlash between the two lead screws amplifies the necessary drive torque.

Servo-drive Systems. The prime movers are often controlled by feedback-error data and to this end nearly all digital-to-analog shaft-position converters show merit.

Incremental Drive Systems. The prime movers may be conventional open-cycle step motors or they may be closed-cycle using incremental pulse feedback to either a conventional prime mover or a step motor. Data must be available in the incremental feedback concerning the direction of the increment. Incremental feedback and incremental pulse direction sensing are discussed in Sec. 18.6.

Strobed Helix Bar. In this type of device a linear helix conventionally rotates at a constant speed, and therefore the strobe timing is a constant function of mark position across the chart. With a straight helix there is a linear relation between the time

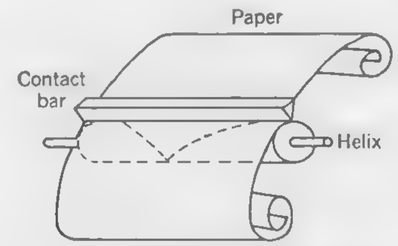


FIG. 18.97. A helix-bar recorder.

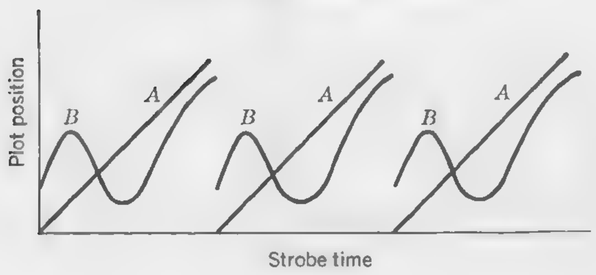


FIG. 18.98.

delay and distance across the chart. However, the helix is not restricted in form so severely. The helix can be so shaped as to give any single-valued functional relationship (the function need not be monotonic) desired for time delay and distance. The frequency response of this type of recorder can be rather excellent, as the value of the variable can be followed if it makes a full-scale change during the period of one sampling time. These recorders, though conventionally used with electrosensitive paper, are not so restricted, for photographic, carbon, and ink-transfer techniques have been employed.

Three or more styli can be equally spaced around a belt which moves perpendicular to the paper motion in such a manner that when one stylus leaves the edge of the paper another one enters at the other edge. Normally, all styli print the same symbol but they can be different, and when this is the case interplexing can yield more than one plot. When used with photographic writing a whirling mirror is often used instead of a spiral.

One of the most important characteristics of a strobed helix-bar recorder is the particular time-delay generator utilized. One system is to place a shaft position analog-to-digital converter on the helix; upon coincidence of the number to be plotted and the shaft-position decoder the helix bar is caused to print. Another method is to count pulses into the register containing the value to be plotted in such a manner as to count the counter down to 0. When this carry occurs the helix bar is caused to print. It should be noted that this counter could have been counted up to full scale and the print initiated at this point had the helix simply been rotated in the opposite direction. A variation on the above system is to count a register up until its value matches that which is to be plotted and at this point initiate the print.

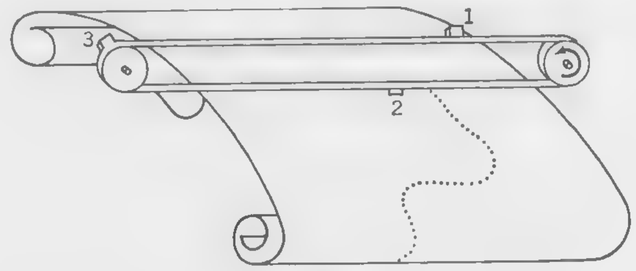


FIG. 18.99.

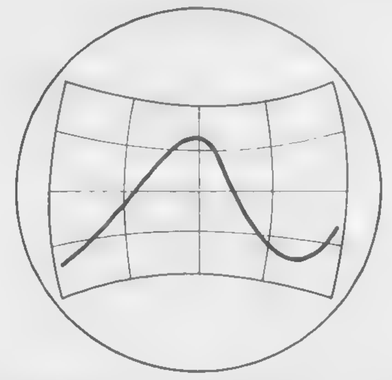


FIG. 18.100. Electronically formed grid lines on a cathode-ray-tube plot.

18.8.3. Cathode-ray Techniques. In these devices the drive method encompasses all the digital-value-to-analog-voltage converters and the driving and linearizing circuitry is that common to the cathode-ray art.

Although it is common to have the grid line embossed in a fixed plastic window through which the cathode-ray tube is viewed, it is possible to generate these grids electronically and to have them pass through all the circuitry contributing to the distortion of the presentation. Therefore, the graph when considered with respect to the grid line is distortion-free. This allows curves to be produced containing inaccuracies of less than the maximum linearity presently obtainable in special cathode-ray tubes (0.1 per cent). Special techniques come to play in the recording of these data. It is necessary to use a shuttered optical recording system for it takes time to draw the curve and care must be exercised to select the proper phosphor for the recordings being made.

Photographic Recording. All the techniques common to cathode-ray photography are applicable here, with the added condition that the camera be controlled by the machine. It is common to allow simultaneous viewing and recording of the plot. This is sometimes done with the aid of two cathode-ray tubes in parallel, or a half-silvered mirror or the newer dichroic techniques. (A dichroic mirror is one in which nearly perfect reflection exists for certain wavelengths of light and nearly perfect

transmission exists for others.) One of the major advantages of the photographic recording of cathode-ray-tube plots is the vast amount of knowledge and experience available to the designer.

Xerographic Techniques. These methods are described in Art. 18.8.1. Their main advantage over the normal photographic process is the decrease in processing time, the completely dry operation, and the lower cost due to the use of ordinary paper.

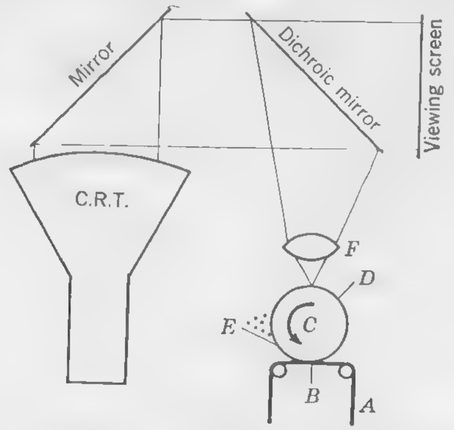


FIG. 18.101. Xerographic recording of cathode-ray-tube plot while viewing through a dichroic mirror: *A*, paper; *B*, paper-charge transfer bar; *C*, dielectric drum; *D*, drum-charging brush; *E*, dry-ink feed; *F*, lens.

This short processing time can be of the order of seconds, and in these instances viewing of the recorded trace is not very different from viewing the cathode-ray-tube screen.

References

1. Blumenthal, E., and F. Lopez: Punched Card to Magnetic Tape Converter for Univac, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 8-11.
2. Brustman, J. A., K. L. Chien, and D. Flechtner: Input and Output Devices of the RCA BIZMAC System, *IRE Convention Record*, part 4, New York, 1956, pp. 88–93.
3. Burke, H. E.: Survey of Analogue-to-digital Data Converters, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 98–105.
4. Byrd, D. J. P., and B. J. Welby: The Input-output Equipment of the Ferranti Digital Computers, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 126–132.
5. DiGiulio, E. M.: Burroughs G-101 High Speed Printer, *IRE Convention Record*, part 4, New York, 1956, pp. 94–100.
6. Eldredge, K. R., F. J. Kamphoefner, and P. H. Wendt: Automatic Input for Business Data-processing Systems, *Proc. Eastern Joint Computer Conf.*, New York, T-92, Dec. 10–12, 1956, pp. 69–72.
7. Epstein, H.: The Electrographic Recording Technique, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 116–118.
8. Epstein, H., and F. Innes: The Electrographic Recording Technique, *IRE Convention Record*, part 4, New York, 1955, pp. 135–138.
9. Epstein, H., and P. Kintner: The Burroughs Electrographic Printer-plotter, *Proc. Eastern Joint Computer Conf.*, New York, T-92, Dec. 10–12, 1956, pp. 73–80.
10. Gray, Walter: RAYDAC Input-output Systems, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 70–76.
11. Greenwald, Sidney: SEAC Input-output System, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 31–36.

12. Gordon, B. M., M. A. Meyer, and R. N. Nicola: Shaft-to-digital Encoder, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 128–133.
13. Hosken, J.: Survey of Mechanical Type Printers, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 106–112.
14. Leiner, Alan L.: Buffering between Input-output and the Computer, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 22–30.
15. Libaw, W. H., and L. J. Craig: A Photoelectric Decimal-coded Shaft Digitizer, *IRE Trans. on Electronic Computers*, **EC-2**(3): 1–4 (September, 1953).
16. Masterson, E., and A. Pressman: A Self Checking High Speed Printer, *Proc. Eastern Joint Computer Conf.*, Philadelphia, Dec. 8–10, 1954, pp. 22–29.
17. Masterson, E., and L. D. Wilson: Output Devices, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 58–61.
18. Mitchell, James M.: High-speed Multichannel Analog-digital Converter, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 118–127.
19. Ogle, James: Optical Display for Data-handling System Output, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 230–232.
20. Pike, James L.: Input-output Devices Used with SEAC, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 36–38.
21. Poppe, C. W., and P. J. Suhr: An Automatic Voice Readout System, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 219–221.
22. Randall, James H.: A Method of Coupling a Small Computer to Input-output Devices without Extensive Buffers, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 136–138.
23. Retzinger, L. P., Jr.: An Input-output System for a Digital Control Computer, *Proc. WESCON Computer Sessions*, Los Angeles, Aug. 25–27, 1954, pp. 67–76.
24. Rosen, Leo: High Speed Printing Equipment, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 95–97.
25. Rossheim, Lt. R. J.: Survey of Nonmechanical Type Printers, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 113–117.
26. Scarbrough, A. D.: An Analog-to-digital Converter, *IRE Trans. on Electronic Computers*, **EC-2**(3): 5–7 (September, 1953).
27. Seehof, J., M. Armstrong, G. Farley, M. Leinberger, M. Markakis, and S. Smithberg: The National Cash Register High-speed Magnetic Printer, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 243–250.
28. Shockency, W. S.: Multi-channel Analog-digital Conversion System for D-C Voltages, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 113–117.
29. Sims, J. C., Jr.: Magnetic Reproducer and Printer, *Proc. Western Computer Conf.*, Los Angeles, Feb. 4–6, 1953, pp. 160–166.
30. Smith, H. M.: The Typotron, A Novel Character Display Storage Tube, *IRE Convention Record*, part 4, New York, 1955, pp. 129–134.
31. Speller, Jack B.: A Digital Converter, *Proc. WESCON Computer Sessions*, Los Angeles, Aug. 25–27, 1954, pp. 29–31.
32. Stevens, L. D.: Engineering Organization of Input and Output for the IBM 701 Electronic Data Processing Machine, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec., 10–12, 1952, pp. 81–85.
33. Tholstrup, H. L.: Perforated Storage Media, *Elec. Mfg.*, **62**(6): 53–61 (December, 1958).
34. Thompson, Russel G., and Clayton E. Hunt: The Eastman Kodak Multiple-Stylus Electronic Printer, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 118–122.
35. Vanderburgh, A.: The Lincoln Keyboard—A Typewriter Keyboard Designed for Computers Input Flexibility, *Commun. Assoc. Computing Machinery*, **1**(7): 4 (July, 1958).
36. Walton, C. A.: A Direct-reading Printed-circuit Commutator for Analog-to-digital Data Conversion, *IBM J. Research Development*, **2**(3): 178–192 (July, 1958).

37. Wilson, L. D., and E. Roggenstein: Input Devices, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S-53, Dec. 10–12, 1952, pp. 53–58.
38. Wilson, Louis D., and Saul Meyer: The Model II Unityper, *IRE Trans. on Electronic Computers*, **EC-2**(4): 19–27 (December, 1953).
39. Winter, A. J.: A Magnetically Coupled Low-cost High-speed Shaft Position Digitizer, *Proc. Western Joint Computer Conf.*, Los Angeles, Feb. 4–6, 1953, pp. 203–207.
40. Zweizig, J. R.: A Digital Voltage Encoder, *IRE Trans. on Electronic Computers*, **EC-3** (3): 25–28 (September, 1954).

Section 19

SPECIAL-PURPOSE COMPUTERS

By

ROBERT R. JOHNSON, *Manager—Engineering, Computer Department, General Electric Company, Phoenix, Ariz.*

MAX PALEVSKY, *President, Packard-Bell Computer Company, Los Angeles, Calif.*

CONTENTS

19.1. Special-purpose Digital Computers 19–1
19.2. Computer Descriptions 19–2
19.2.1. Simultaneous Linear Algebraic Equations 19–2
19.2.2. Partial Differential Equations 19–4
19.2.3. Logic Computers 19–5
19.2.4. General Mathematical Operations 19–9
19.2.5. Digital Techniques 19–12
19.3. The Digital Differential Analyzer 19–14
19.3.1. Introduction 19–14
19.3.2. Principles of Operation 19–15
19.3.3. Generation of Special Functions 19–29
19.3.4. Error 19–42
19.3.5. Input and Output 19–53
19.3.6. Over-all Computer Design 19–59
19.3.7. Programming 19–61
19.3.8. The Synthesis of General-purpose Computer and Digital Differential Analyzer Techniques 19–67

19.1. SPECIAL-PURPOSE DIGITAL COMPUTERS*

A computer is a machine designed to carry out calculations or to perform specified transformations of information. Mechanized data processing, interpretation, and computation are an expanding art. Large-scale computers have placed a general emphasis on numerical methods. This emphasis stems in part from an interest in greater accuracy in solving more complex systems. This interest has led to applications of digital techniques to problems having special characteristics. A special-purpose digital computer is defined here to be a computer which *can only* be used for certain special problems; the methods of solution for such computers must involve the processing of discrete numbers. This definition does not include computers which have certain features adapting them to special classes of problems if the general applicability of the computer is not limited.

The basic reason for the appearance of special-purpose digital computers is that these machines often satisfy a need for economic solutions to a given class of problems. There are many types of mathematical operations for which the large general-purpose

* By Robert R. Johnson.

computers are not very efficient. The examples to follow illustrate certain cases for which such special computers have been designed. Another uneconomical aspect of the large general-purpose machines is the programming requisite to any problem solution. Many special-purpose computers are designed to accept a class of problems in its normal form and produce solutions in a form familiar to the scientist. It is often the case that the special machine can generate solutions much more rapidly and with far less equipment than the general-purpose machine. In addition, the small special computer can be made available to many more scientists, and it can relieve the large and versatile computers of many typically tedious and commonplace problems.

On the other hand, the special-purpose computer often depends on a peculiar feature of its problem for an economic circuit implementation. This means that such machines usually require a large amount of special design and engineering, both quite expensive. Also, there are many applications requiring solutions to several special types of problems.[34] One general-purpose computer with certain features could provide a much more economic solution to the over-all application than could several small special computers. Another objection to special computers is that it often develops that a computer must be built before the problem characteristics are known.

Following the paths of evolution, however, many problems have arisen demanding special computers for their solution. The principles of problem formulation, logical design, and the methods employed in solving these problems could be of value to the designers of general-purpose and special-purpose digital computers.

19.2. COMPUTER DESCRIPTIONS*

19.2.1. Simultaneous Linear Algebraic Equations. Problems in all branches of science lead to systems of simultaneous linear algebraic equations. Many mathematical techniques have been devised to obtain solutions to systems ranging from a few to infinitely many independent variables. General-purpose digital computers are commonly programmed to solve such systems of equations, but these machines are expensive, difficult to program, and slow in their effective operation. Since problems involving many linear equations arise so often, at least two special-purpose digital computers have been developed to generate their solutions. Haller, Raymond & Brown Co. has manufactured a computer capable of handling systems involving up to 1,200 equations with a possible accuracy of 1 part in 10,000.[69] Fairchild Engine and Airplane Co. has constructed another computer under contract with the United States Air Force. This computer can handle up to 300 equations and carries 4 decimal digits per word.[66]

Both computers utilize the Gauss-Seidel iteration. Other more elegant iteration methods can produce solutions more rapidly, but the advantage of the Gauss-Seidel technique is its computational simplicity. Writing the system of equations

$$\begin{aligned} a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n &= b_1 \\ a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n &= b_2 \\ a_{n1}X_1 + a_{n2}X_2 + \cdots + a_{nn}X_n &= b_n \end{aligned}$$

the first step in the solution is assuming any set of initial values for the x_i. The values

$$X_1^{(1)} = \frac{b_1}{a_{11}} \qquad X_2^{(1)} = \frac{b_2}{a_{22}} \qquad \cdots \qquad X_n^{(1)} = \frac{b_n}{a_{nn}}$$

are suggested as the initial approximation. Using these assumed values, the first residual is computed:

$$R_1^{(1)} = a_{11}X_1^{(1)} + \cdots ; + a_{1n}X_n^{(1)} - b_1$$

* By Robert R. Johnson.

The improved approximation $X_1^{(2)}$ is found:

$$X_1^{(2)} = X_1^{(1)} - \frac{R_1(1)}{a_{11}}$$

Using $X_1^{(2)}$ the second residual is computed and used to find $X_2^{(2)}$:

$$R_2^{(1)} = a_{21}X_1^{(2)} + a_{12}X_2^{(1)} + \cdots + a_{1n}X^{(n)} - b_1$$

$$X_2^{(2)} = X_2^{(1)} - \frac{R_2}{a_{22}}$$

This process is continued through all n variables using the newest values for each $X_q^{(j)}$; repeating this entire sequence for j iterations produces a continually improving approximation to the exact solutions X_q. The values of $R_q{}^j$ are a measure of the accuracy of the approximations $X_q^{(j)}$. As pointed out in Whittaker & Robinson,[71] iteration techniques such as this one have the advantage that calculation errors continually correct themselves in subsequent steps of the computation.

Haller, Raymond & Brown Computer. The binary computer made by Haller, Raymond & Brown, Inc., uses a magnetic-tape loop to store the coefficients a_{pq}, and a magnetic drum for random access to store the values $X_q^{(j)}$. Conventional hard-tube flip-flops and gating circuits are used to implement the logical design. Input is via a keyboard entry device with outputs being programmed out into a display panel. Approximately 400 individual plug-in units constitute the arithmetic and control equipment, with most of the arithmetic operations being performed in parallel. Multiplication is by repeated addition.

The logical design of this computer is arranged to save computation time by omitting all vanishing coefficients a_{pq}. The nonvanishing coefficients are entered on the tape loop along with adequate coding information to indicate the type of coefficient (a_{pp},a_{pq},b_p) and the column index q. The unknowns $X_q^{(j)}$ are stored serially on the drum in 19 tracks of 64 words having 25 bits per word. The column index X_q acts as the drum-memory-address information for each unknown.

The computational procedure is simple, and it is more or less fixed by the machine structure. Hence very little programming problem-preparation time is required. The matrix coefficients are read sequentially into the arithmetic unit from the tape. Each residual is computed using part of the drum as "scratch-pad" storage. From this the new value for each unknown is computed. The computation of the residuals and of the new values for the unknowns is controlled by a simple programmed routine stored on the drum.

The machine is in use at Haller, Raymond & Brown, Inc. Its operational speed is indicated by noting that 3 min per iteration are required for a 73-equation problem while 20 min per iteration is the required computing speed for a 793-equation problem. An improved version of this same computer was completed in September, 1955. Circuit and component simplicity and reliability are the major features of the improved model. Computation services are available from the manufacturer.

USAF-Fairchild Computer. The USAF-Fairchild computer is a special-purpose decimal digital computer designed to handle simultaneous equations involving up to 300 variables. Two synchronous magnetic tapes are used for storage. A long tape loop holds the coefficients and their associated orders, while a short tape loop holds the unknowns. The computational principle utilized by the computer is the Gauss-Seidel iteration. The matrix coefficients and their corresponding unknowns are read simultaneously from the tapes. No special treatment is afforded zero coefficients. Each residual is accumulated and the improvement in the corresponding unknown is computed in sequence. Improved values in the unknowns are used in computations for subsequent residuals.

Input to the computer is performed with a keyboard, and the output is printed. Approximately 2,000 tubes constitute the electronic equipment of the machine. The arithmetic operations are carried out with a multiplier, accumulator, a temporary storage register, and the necessary control equipment. The multiplier uses excess-three

binary coded digits and a diode matrix which generates all possible products from this configuration. The 16 sets of partial products obtained from multiplying two 4-digit numbers are accumulated successively from each multiplication used in computing the modified residuals. No division is required since the initial system of equations has each row divided through by a_{ii}; hence the modified unknowns are computed:

$$X_q^{(j+1)} = X_q^{(j)} - R_2^{(j)}$$

where

$$R_q^{(j)} = X_1^{(j)} + \frac{a_{12}}{a_{11}} X_2^{(j)} + \cdots + \frac{a_{1n}}{a_{11}} X_n^{(j)} - \frac{b_q}{a_{11}}$$

The temporary storage register is used to store the new values $X^{(j+1)}$ until the short tape returns to the properly synchronized location.

The computer was completed in February, 1950, and was placed in operation at the Oak Ridge National Laboratory in 1951. Subsequently, the machine was dismantled, and many of its features have been included in the ORACLE computer.

19.2.2. Partial Differential Equations. *DINA.* The solutions of many problems in mathematical physics are represented by the solutions of the following four partial differential equations:

Laplace: $\nabla^2 u = 0$

Poisson: $\nabla^2 u = p$

Diffusion: $\nabla^2 u = k^2 \dfrac{\partial u}{\partial t}$

Wave: $\nabla^2 u = \dfrac{1}{v^2} \dfrac{\partial^2 u}{\partial t^2}$

Solutions for these equations are uniquely specified in terms of the equation and a corresponding set of initial and boundary conditions.[16,63]

Many special analog techniques have been developed to describe the behavior of physical systems governed by these and other sets of partial differential equations. To obtain more accuracy and versatility, numerical methods are often used to obtain solutions for many such equations. The partial differential equations are generally replaced by some form of finite-difference equation to facilitate the numerical work. These differences are usually based on the values of the function at a set of points selected in any prescribed manner in the region of interest.

A special-purpose digital computer has been designed to solve any of the four equations listed above.[36] The computer is mechanized to solve the finite-difference equation corresponding to each analytic equation. Using the Richardson iteration,[24] the difference equations for the two-dimensional case are tabulated below.

Laplace:

$$4u^{(m+1)}(x_1,x_2) = u^{(m)}(x_1 + h,x_2) + u^{(m)}(x_1,x_2 + h) + u^{(m)}(x_1 - h,x_2) + u^{(m)}(x_1,x_2 - h) = \sum u^{(m)}$$

Poisson: $$4u^{(m+1)}(x_1,x_2) = \sum u^{(m)} - h^2\rho(x_1,x_2)$$

Diffusion: $$\frac{1}{h^2}\left[\sum u^{(m)} - 4u^{(m)}(x_1,x_2)\right] = \frac{k^2}{\Delta t}\left[u^{(m+1)}(x_1,x_2) - u^{(m)}(x_1,x_2)\right]$$

or $$4u^{(m+1)}(x_1,x_2) = \sum u^{(m)} \qquad \text{for } \Delta t = \frac{h^2k^2}{4}$$

Wave: $$2u^{(m+1)}(x_1,x_2) = \sum u^{(m)} - u^{(m-1)}(x) \qquad \text{for } h^2 = 2v^2(\Delta t)^2$$

This is for a square lattice with interval $h = \Delta x_i$ with the function n computed at the coordinates x_i for the mth application of the iteration. No attempt is made to use general relaxation methods.

Since a three-dimensional machine is a straightforward extension, the two-dimensional computer is designed to handle any of the four basic equations in terms of two coordinates. A magnetic drum is proposed in the reference to store a table containing the values of the function at each point of a 100 × 100 lattice. Using serial binary numbers of about 20 bits each, this requires storage for 2×10^5 bits for the table.

The computational principle used in solving these equations is the repeated application of the difference equation. The initial input to the table introduces the boundary values, the initial conditions, and the assumed solution value at each lattice point. The computation proceeds starting in one corner of the table by recomputing the value of the function with the Richardson method for the Laplace equation. This gives the function at one point as the average value of the function at each of the surrounding four points. Using this principle, the computer traverses the lattice repeatedly until a solution of sufficient accuracy is attained. The convergence rate of these computations is considered by Frankel.[24]

The computational equipment varies somewhat for each equation. The technique is simply the evaluation of the finite-difference equation. The equipment required to perform this computation is determined by the terms present in each equation. Thus for the Laplace and diffusion equations, two extra storage channels are required to retain the old values $u^{(m)}(x_1 - h, x_2)$ and $u^{(m)}(x_1, x_2 - h)$. In addition, for the Poisson equation, a second complete table representing all 10^4 lattice points is used in storing $\rho(x_1,x_2)$. Similarly, a complete two-dimensional table is required for storing $u^{(m-1)}(x_1,x_2)$ for the wave equation. In each case the proper terms are read from the memory shifted by the appropriate number of binary places to perform the necessary multiplications by constants selected to be integral powers of 2, added together and replaced in the memory tables.

The equipment proposed to implement this machine is summarized as follows:

1. Input-output unit
2. Two complete two-dimensional storage tables; one two-column storage table, both approximately 4×10^5 bits
3. Selection matrices
4. Four serial binary adders
5. Four temporary storage channels
6. Timing channels

The advantage of a machine of this type is that it is cheaper to build, easier to operate, and generates solutions of these equations considerably faster than do most general-purpose computers. A disadvantage is that it still is not nearly so fast or easy to use as many of the analog methods, even though the digital technique provides greater inherent accuracy.

19.2.3. Logic Computers. The fundamentals of modern logical science were established by George Boole with the publication of *The Mathematical Analysis of Logic* in 1847,[5] and *Of the Laws of Thought* in 1854.[6] W. S. Jevons contributed the concept of solving logical problems by eliminating classes of variables.[31] The first machine for implementing the solutions of logical systems was constructed by Jevons in 1869.[30] This computer is in the Museum of the History of Science, Oxford, England.

Jevons Machine. This first logic computer handled four variables. The heart of the machine was Jevons logical abacus.[31] This consisted of a framework holding wooden markers; each marker represented one of the 16 possible minterms of the four variables. A keyboard was attached to the framework by various mechanical linkages. A set of keys representing each variable and its complement was provided for the predicate and another for the subject of the logical proposition. The linkages were constructed to select only these combinations consistent with the rules of the proposition inserted into the machine. Another set of keys denoted the logical relationships which were provided in the device.[7] The computer output was visual inspection of the markers indicated in the abacus framework.

A. Marquand designed an improved four-variable computer around 1885 which utilized levers, rods, and wires.[43] He appears to have been the first to propose and design an electrical device to solve logical equations.[44]

Kalin-Burkhart Calculator. The Kalin-Burkhart logical truth calculator is the first electrical implementation of the truth-table method for solving logical relationships.[8]

The computer handles 12 logical variables and scans the 4,096 possible combinations sequentially. Multiple-position switches set up the variables and specify the rules connecting the terms. The five internal operations are: complementation, conjunction, inclusive disjunction, conditional implication, and biconditional implication. The computer examines one line of the truth table at a time. All combinations consistent with the rules are designated with lamps. Using relays to scan all lines of the truth table, the computer requires about 38 min to evaluate all the possible combinations of the 12 variables connected with the five rules above.

Ferranti, Ltd., constructed a seven-variable relay machine to scan sequentially the 128 possible combinations in the truth table.[7] The added feature over the Kalin-Burkhart machine is the inclusion of the exclusive disjunction in the available connectives. Input is via a plugboard, and output is indicated by lights or by stopping the scanning mechanism.

IBM Decision Machine. Dr. M. E. Maron of the International Business Machines Corp. constructed a logic computer for eight variables using essentially standard IBM relays, plugboards, and switches. The machine uses the five logical operations of the Kalin-Burkhart device. In addition, it solves for one of three classes of propositions:

1. Those which are factually true or false
2. Those which are tautologically true
3. Those which are logically false (self-contradictory)

These are distinguished by the truth-table values of the propositions. Factually true sentences are true (or false) only for certain values of the variables; tautological propositions are either true or false for all values of the input variables.

The decision machine constructed provides 14 inclusive disjunction circuits, 14 conjunction circuits, 10 complementation circuits, 6 conditional circuits, and 4 biconditional circuits. Each variable is represented by two 4-point relays; thus each variable appears on seven common, seven normally open, and seven normally closed points. A plugboard is used to connect the variables through the specified connectives to a final output relay.

The conventional methods of representing logical rules with relay circuitry are used in the computer.[57] Complements are considered the normally closed relay contacts while the variables are represented by the normally open contacts. Conjunctions are specified by connecting contacts in series and disjunctions are contacts connected in parallel. Conditionals are implemented using a disjunction:

Conditional: $A \supset B$ is equivalent to $\bar{A} + B$.

Biconditional: $A \equiv B$ is equivalent to $A \cdot B + \bar{A}\bar{B}$.

Each complete operation is prewired onto the plugboard. In preparing a problem for the computer the appropriate variables are interconnected using these standard operations. The output lead from the specific proposition is connected to the output relay; a true value for the statement closes the output relay.

The totality of combinations are scanned by connecting an eight-stage binary counter to the input logical variables. The input to each variable corresponds to one counter stage. Supplying 2^8 pulses to the counter cycles all eight variables through all possible combinations. Lights indicate true and false combinations. To determine tautological truth or falsity, two latching relays are used. These relays close if there are one or more false or true combinations, respectively, in the truth table. Approximately 50 sec suffice to inspect the 256 possible combinations of the eight variables.

Ferranti. A different type of truth-table computer was built by Ferranti in 1951 to illustrate the feedback principle. Four variables can be manipulated by the machine. Twenty relays are interconnected to set up the proposition and scan the truth table. Combinations which fail to satisfy the proposition generate feedback signals which are used to change the state of the input variables. A satisfactory combination of input variables terminates the cycle. On relatively simple problems the feedback machine can be many times faster than the scanning-type computer. Both the Ferranti computers are described in some detail in the literature.[45]

Vendac. Electronic digital computers for handling logical problems were constructed as early as 1950.[2] The Vendac, at the National Physical Laboratory, Pretoria, was designed to represent the Venn-diagram concept by means of a pulse-code technique. Four classes of variables and their complements are represented by individual pulse trains. Each pulse train is coded so as to present a different time sequence of pulses. The respective pulse trains are combined in coincidence, anticoincidence, and mixer circuits to obtain the truth value of the equation represented by the patchcord interconnections among the gating units. The output is fed either to true-false jacks or to an oscilloscope for direct viewing. The output-pulse pattern yields these values of the input variables which satisfy the equation plugged into the computer.

One unusual feature of this computer is the technique used to handle redundancies. The mixer circuits actually sum the pulse-voltage outputs of the gating circuits so that a redundancy of order n is indicated by a pulse of amplitude nA. Here A is the basic pulse amplitude. Contradictions are handled by using two different pulse amplitudes as outputs from gates. Inequality between two input variables is indicated by a positive output pulse of amplitude A. When the input variables are equal, a negative output pulse of amplitude $A/2$ is generated. A contradiction between two subclasses is indicated by a positive pulse of amplitude $A/2$.

In essence, the computer is composed of a counter, logical operators, and an output device. The "pulse trains" are the time sequence of digits appearing on each flip-flop when the counter is cycled through all possible combinations. Each variable is represented by one stage of the counter. The logical gates are interconnected to operate on the signals from the counter. The outputs are derived from the gating network by conventional circuit techniques. A 100 Kc per sec clock rate is utilized.

Burroughs' Computer Using Polish Notation. A unique electronic computer has been developed by Burroughs Corporation.[11] This machine is designed to evaluate logical formulas expressed in Lukasiewicz parenthesis-free notation.[38] The advantages of this method of computation are:

1. Formulas may be evaluated by one scan from right to left.
2. With some practice, it appears to be as easy to write the Lukasiewicz form directly from a "compacted" Boolean expression as it is to expand it.

However, it is easier to formulate a problem using conventional Boolean techniques. A Boolean expression which has been multiplied out is probably as simple a function to evaluate, even though more terms may be involved. Furthermore, there appear to be no simple basic theorems for use in manually simplifying formulas expressed in Lukasiewicz's notation.

One way to use the Burroughs computer in logical design problems is to provide the algebraic conditions as a set of simultaneous equations. The computer then would provide the various sets of values for the algebraic variables satisfying the set of equations. The user must select the set having the simplest formulation. In most applications the Burroughs machine provides the same functional operation as any of the other truth-value computers.

The method of evaluating a formula expressed in the parenthesis-free notation is illustrated in terms of Boolean expression

$$f_B = ab\bar{c} + \bar{a}(d + e)$$

The notation here is as follows: lower-case letters denote variables, and upper-case letters denote Lukasiewicz logical operators.

$A \equiv$ alternation (inclusive "or")
$K \equiv$ conjunction ("and")
$N \equiv$ complementation ("not")

The Lukasiewicz formulation f_L is written starting from the left side of f_B. The logical operators A and K refer to the two values denoted by the pair of variables following each appearance of the operators. Thus

$a \cdot b$ corresponds to Kab
$a \cdot b \cdot c$ corresponds to $KaKbc$

The operator N changes the state of the variable on its right and does not affect the region of influence of K or A.

$$a \cdot b \cdot \bar{c} \text{ corresponds to } KaKbNc$$

The Boolean formula f_B is rewritten without parentheses as follows:

$$f_L = AKaKbNcKNaAde$$

The computer evaluates this formula by starting on the right end. The values of successive symbols are read and operated upon sequentially. The rank of an equation is determined by the number of variable values that the computer must store in evaluating pairs of variables.

The equation above is of rank 3 and length 13. In operation the computer stores the following numbers.

Computer reads	And stores in memory spots		
	No. 1	No. 2	No. 3
e	e		
d	d	e	
A	$d + e$	d	
a	a	$d + e$	
N	$\bar{a}$	$d + e$	
K	$\bar{a}(d + e)$	$\bar{a}$	
c	c	$\bar{a}(d + e)$	
N	$\bar{c}$	$\bar{a}(d + e)$	
b	b	$\bar{c}$	$\bar{a}(d + e)$
K	$b\ \bar{c}$	b	$\bar{a}(d + e)$
a	a	$b\ \bar{c}$	$\bar{a}(d + e)$
K	$abc-$	a	$\bar{a}(d + e)$
A	$ab\bar{c} + \bar{a}(dte)$	$ab\bar{c}$	$\bar{a}(d + e)$

The functions which the Burroughs truth-function evaluator can handle are listed below.

Variable p	Values q	Complement of p Np	Inclusive disjunction Apq	Exclusive disjunction Rpq	Conjunction Kpq	Biconditional Epq	Conditional Cpq
0	0	1	0	0	0	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	0	0
1	1	0	1	0	1	1	1

The actual computer is set up to handle formulas of length 98 and rank 6 using 10 variables. The formula is plug-wired into the control panel. Two modes of operation are possible. In the first, the computer evaluates the plugged-in formula for a specified set of values of the variables. In the second mode, the evaluator generates in sequence all possible combinations of the 10 variables and tests each combination in the formula. The automatic mode is terminated when the value of the formula satisfies a preset switch setting (0 or 1). This computer is built using pluggable relays and diodes and operates at 10 characters per sec. This speed corresponds to a 2¾-hr time for 10 variables in a formula of length 98.

Bell Telephone Laboratories. The Bell Telephone Laboratories have constructed a relay machine which differs from all the preceding computers.[59] This machine simulates a combinational relay circuit on a plugboard while having a set of switches which are used to set up Boolean formulas. Each switch represents one of the 16 minterms of 4 variables. The unique feature of this machine is that each switch has three positions: each minterm can be true, false, or either. Hence the machine can provide reduced combinational circuits when the presence or absence of certain minterms does not matter. The five modes of operation of the computer are:

1. Evaluate: indicates which minterms satisfy the plugged-in formula
2. Compare: indicates which minterms disagree with the plugged-in formula
3. Short: indicates which contacts of the plugged-in formula can be shorted out
4. Open: indicates which contacts of the plugged-in formula can be opened
5. Prove: indicates which kinds of contacts are necessary to satisfy the switch conditions

19.2.4. General Mathematical Operations. *Polynomial Root Extractor.* A small digital computer has been constructed at the California Institute of Technology to extract the complex roots of high-degree polynomials.[32] This computer is designed to handle polynomials up to the sixteenth degree. It obtains the complex roots of polynomials having complex coefficients with a significance of 19 binary places. Solution times average about 16 sec per root.

The general computational method is that of evaluating a polynomial in the complex plane. The computer is designed to (1) evaluate the polynomial for successive increments in its complex argument, and (2) select the complex increment that always decreases the absolute value of both the real and imaginary components of the polynomial.

The computer generates the value of the polynomial at a given point by computing each value of the polynomial and its derivatives at a series of intermediate points. Each intermediate point is separated by a distance Δ:

$$\Delta = \pm\delta \qquad \text{or} \qquad \Delta = \pm j\delta$$

The value of the nth derivative of an nth degree polynomial is a constant. Starting with this highest derivative, each lower derivative is computed at each intermediate point using a truncated Taylor series.

$$f^{(m)}(Z_0 + \Delta) = f^{(m)}(Z_0) + \Delta f^{(m+1)}(Z_0) \tag{19.1}$$

The computer actually retains and uses the real and the imaginary components of the polynomial $f(Z)$ and its n derivatives. Thus, in computing the value of the polynomial at the point $(Z_0 + \Delta)$, the computer obtains either of two possible sums depending on whether Δ is real or imaginary.

$$f(Z_0)u(Z_0) + jV(Z_0)$$

For $\Delta = \pm\delta$:

$$\begin{aligned} u(Z_0 + \Delta) &= u(Z_0) \pm \delta u^1(Z_0) \\ v(Z_0 + \Delta) &= v(Z_0) \pm \delta V^1(Z_0) \end{aligned} \tag{19.2}$$

For $\Delta = \pm j\delta$:

$$\begin{aligned} u(Z_0 + \Delta) &= u(Z_0) \mp \delta v^1(Z_0) \\ v(Z_0 + \Delta) &= v(Z_0) \pm \delta u^1(Z_0) \end{aligned}$$

Starting with the nth derivative at Z_0, the computer evaluates the real and imaginary components of each lower derivative at $(Z_0 + \Delta)$ using this formula.

The second design objective is to select a particular value of Δ from (19.1) such that

$$\begin{aligned} |u(Z_0 + \Delta)| &\leq |u(Z_0)| \\ \text{and} \qquad |v(Z_0 + \Delta)| &\leq |v(Z_0)| \end{aligned} \tag{19.3}$$

The computer examines the 16 possible combinations of sign which occur in (19.3) when computing $u(Z_0 + \Delta)$ and $V(Z_0 + \Delta)$ from $u(Z_0)$, $u^1(Z_0)$, $v(Z_0)$, and $v^1(Z_0)$. It can be shown that one of the directions from (19.1) can always satisfy the requirements (19.4).

At each intermediate point, the computer selects this necessary Δ and then evaluates the n derivatives of the polynomial before calculating the value of the polynomial by (19.3).

Any polynomial is prepared for the computer in three simple steps. The first is to transform the polynomial to a new variable w such that all roots of $f(w)$ are enclosed in the unit circle in the w plane.

$$w = \frac{Z}{R} \tag{19.4}$$

where $R > \left|\frac{a_k}{a_n}\right| - 1$ and (a_k) is the largest coefficient in the polynomial

$$f(Z) = a_n Z^n + a_{n-1} Z^{n-1} + a_1 Z + a_0$$

The second step is to divide $f(w)$ by the quantity $(2n{:}A_n)$ where A_n is the coefficient of w^n in the transformed polynomial. This is to ensure that the absolute values of the derivatives never exceed unity when the computer is searching for roots. The third step is evaluating the real and imaginary parts of the polynomial and its n derivatives at $w = 0$. These quantities are entered into the computer as initial values. These initial values form the basis of the first direction decisions and computations, Eqs. (19.1) and (19.3), respectively.

This method of computation was chosen because the associated computer is quite simple. One recirculating memory, 20 flip-flops, and about 200 germanium diodes are the principal computer elements. The memory has 19 word positions; each word consists of 62 binary-digit positions with 2 bit positions separating each word. The word structure is such that each word holds a derivative. One word contains the value of w which is the independent variable, and a second additional word holds a marker corresponding to Δ. Within each word, the values of the real and imaginary components of a derivative are represented as biplexed binary numbers.

One of the basic features of the computer is the utilization of two values of δ. A large value of $\delta = (2^{-10})$ is used with an improved approximation formula:

$$f(Z_0 + \Delta) = f(Z_0) + \frac{\Delta}{2}[f^1(Z_0) + f^1(Z_0 + \Delta)] \tag{19.5}$$

A small value of $\delta = (2^{-20})$ is used with the approximation (19.2). This feature provides rapid root locations without impairing the accuracy required to refine the root location with a small value of δ.

The method employed to locate roots other than the one nearest the origin was selected using the criterion of computer simplicity. The computer has no internal programming for its normal add-subtract routine so that it would be very difficult to divide out roots as they are located. The method used is to continue the normal computation cycle but to force a desired Δ selection. The operator chooses the direction in which he wishes to look for roots and forces the computer to move in that direction. When the operator releases it, the computer seeks the root nearest the release point. This technique allows the operator to look for roots in areas of interest, and it also allows him to find just one of each pair of conjugate complex roots.

Digital Correlator. One of the mathematically useful concepts involved in studying stationary statistical phenomena is that of the correlation function.[72] In many engineering problems, it is desirable to obtain correlation-function information about certain randomly varying signals. Many analog computers have been proposed to generate the autocorrelation and cross-correlation functions of experimentally available data. An electronic digital correlator built at MIT obtains these correlation functions for electrical signals covering the broad frequency spectrum 0 to 12 Mc per sec.[61]

The basic principle of the computer is to approximate the correlation function

$$\phi(\tau) = \lim_{T \to \infty} \frac{i}{2T} \int_{-T}^{T} f(t) f(t + \tau)\, dt$$

by averaging a large number of samples taken at discrete time intervals t_i:

$$\phi(\tau) \doteq \frac{1}{N} \sum_{i=1}^{N} f(t_i) f(t_i + \tau)$$

The two input functions to the computer are electrical signals whose amplitudes are sampled periodically. The autocorrelation function is obtained if both inputs are the same, while the cross-correlation function is obtained if different input signals are used. Increments in the delay interval τ are available in steps of 0.1 μsec ranging from 1 μsec to about 0.1 sec. The computer obtains the quantity $N\phi(\tau)$ where N may be any number of samples; this number is determined by the length of time the computer is allowed to operate with any single value of τ.

The sum of the N products $f(t_i)f(t_i + \tau)$ is obtained by first sampling the amplitude of one input signal with a narrow pulse and converting this voltage level to a binary number. After an elapsed time τ the second input signal amplitude is sampled similarly and converted to a binary number. These two numbers are multiplied in a serial binary multiplier. The products of successive pairs of these samples are accumulated in another register. The computer outputs available are graphical plots of the correlation functions as well as direct digital record of the sums.

The logical design of this computer is implemented as follows: The input voltages are sampled using a sharp pulse from a blocking oscillator. This pulse is applied to the suppressor grid of a gate tube. A capacitor is charged to the sampled voltage and held at this value until the corresponding binary number has been generated. The number generator consists of a sawtooth generator and a gated fixed-frequency pulse generator. The sampling pulse initiates the sawtooth waveform. It also opens a gate supplying timing pulses to a binary counter. Equality between the sawtooth amplitude and the amplitude of the sampled waveform terminates the gate supplying the timing pulses. A 5-Mc clock rate is used for these timing pips. Each pair of numbers is placed in a 10-bit flip-flop register. A 10-bit accumulator sums the partial products of the two numbers representing $f(t_i)$ and $f(t_i + \tau)$. The additions are performed serially using an "exploring pulse." For multiplication, the additions of the multiplicand into the partial product are gated by the successive bits of the multiplier. Shift pulses are used to shift the multiplicand to the left and the multiplier to the right. The product of each pair of samples constitutes one of the N sets of values corresponding to t_i. The accumulated sum of all N pairs of samples represents N times the correlation function $\phi(\tau)$.

Number Sieve. One of the basic problems in number theory is obtaining solutions for linear congruences. By definition, if $a - b$ is divisible by $m > 0$, one says that a is congruent to b modulo m. Thus if a and b are congruent, mod m, one obtains the same remainders in dividing a or b by m.

Professor D. H. Lehmer at the University of California has designed a special-purpose electronic digital-number sieve. This machine is used in finding numbers such that each number will have a specified remainder when divided by any or all of the first 33 prime numbers. The input to the computer is the acceptable remainders corresponding to each prime. Thus possible remainders are:

Remainder of	*Modulo*
1	2
0,2	3
0,1,2	5
.	.
.	.
.	.
1,31,62,85	137

The machine scans all natural positive integers until it finds a number satisfying the imposed conditions.

The computer consists of 33 ring counters. A gating circuit indicates coincidence between the remainders specified with the input switches and the remainders existing

in the counters. Each ring counter has a different prime number of stages. The gating circuit on each ring simply indicates when a particular stage is present. Consider finding the first number that has a remainder of 2 when divided by 5, a remainder of 2 when divided by 3, and a remainder of 1 when divided by 2. The gates on the 5, 3, and 2 ring counters are set to register when the count pulse is in the second and first stage of these respective counters. Preset all counters to 0, and then step all the counters simultaneously. During the seventeenth count, each of the gates on the 5, 3, and 2 ring counter will open and stop the computer. The number 17 is the smallest number satisfying the remainder criterion above. This number 17 is read from the ring counters themselves, although a separate counter could be used for this purpose.

The computer can commence counting with any preset number. Any or all combinations of remainders can be entered into the computer. And in principle, the computer could be extended to find congruences modulo any integer. The counting speed of the machine is 100,000 counts per second.

19.2.5. Digital Techniques. *A Digital Electronic Musical Instrument.* This instrument[67] simulates predetermined sound waveforms with the aid of standard Eccles-Jordan flip-flop circuits, voltage-level gating, and a digital-to-analog converter. The instrument, while actually capable of reproducing the periodic waveform of any sound at any audio frequency, was designed principally for musical sounds. The different sounds are "plugged" into the instrument in the form of logical "Boolean" equations on a plugboard located on top of the instrument.

The mechanism of the instruments operation is as follows: A particular sound waveform is "quantized" (that is, with step functions) to any desired degree of accuracy in the time domain (n intervals along the ordinate and m along the abscissa). These quantized levels are then given numerical values from 1 to n according to their position along the ordinate. Logical equations for a mod n counter are then written which describe, in binary code, the sequence of step functions. Next the equations are plugged into the instrument.

The internal circuitry of the instrument converts the equations into diode voltage-level gating such that the flip-flops will count through the predetermined sequence. A precision-resistor ladder network converts binary numbers represented by the flip-flop counter outputs to an analog waveform identical to the original quantized waveform. This resulting waveform is then electronically smoothed and fed into a high-fidelity audio amplifier and speaker.

The usefulness of this instrument lies in its great accuracy of reproduction of known sounds. This is done without the necessity of any analysis of the sounds in the frequency domain.

Pulse-height Analyzers. In many types of physical research, data in the form of voltage or current pulses appear in large numbers. For example, in bombarding atomic particles with electrons or neutrons a large number of fragmentary particles may be split off. It is often desirable to count these fragments and sort them according to their energy levels. Many analog and digital analyzers have been built to count and sort the voltage pulses generated by the detectors used in these nuclear experiments.

At the California Institute of Technology a simple magnetic-drum digital pulse-height sorter was designed by Dr. S. P. Frankel. This sorter is unique in the sense of its ultimate simplicity. Two recirculating memory tracks are used on the drum. The short track contains a number of binary-digit positions; each bit position corresponds to one pulse-voltage level. A 1 indicates that a pulse of that magnitude occurred; a 0 indicates no pulse at that voltage level. The long track has a number of word positions corresponding to the number of voltage levels used in the short track. Each word in the long track contains the binary total indicating the number of pulses of that voltage level accumulated during the experiment.

A linear sawtooth voltage is synchronized with the recirculation time of the short register. During each sweep the amplitudes of the incoming voltage pulses are compared with the amplitude of the sawtooth. Amplitude coincidence triggers a flip-flop and places a 1 in the corresponding bit position of the short register. By arranging the length of the short register so that it precesses one bit position per word read from the long track, a simple one-bit adder adds each counted pulse of a specific voltage to the totals held in the corresponding word on the long track.

A commercially available pulse-height analyzer is being marketed by the Pacific Electro-Nuclear Co. in Culver City, Calif. This instrument quantizes the amplitudes of pulses derived from any detector into 100 discrete levels or channels. It stores the counts obtained on all channels until the desired spectral statistics are obtained. A printer ordinarily tabulates the totals accumulated in each channel.

The amplitude of each input pulse is sampled and adjusted to a preset threshold level. A phantastron generates a sawtooth voltage whose amplitude is compared with the input-pulse amplitude. The memory-channel address corresponding to the input pulse is derived by gating a string of oscillator pulses into an accumulator. The gate is opened at the start of the sawtooth and closed by the amplitude coincidence. The channel address is thus a number proportional to the pulse amplitude. This number addresses a particular channel in the main memory, and the instrument adds one to the previous contents of the channel. This operation is repeated for each input pulse. The memory consists of rectangular hysteresis-loop magnetic cores mounted in twenty 10×10 matrix planes.

The resolving power of the Pacific electro-nuclear analyzer is specified as follows: Input-pulse amplitudes are quantized into either ½- or 1-volt levels, ± 0.1 per cent. Each input pulse requires $4n + 13$ μsec for conversion and storage. Here n is an integer ranging from 1 to 100 corresponding to the channel number (pulse-voltage amplitude level). For a balanced spectrum, this rate corresponds to 4,500 counts per second; the maximum counting rate is 30,000 counts per second.

Other basically digital pulse-height sorters have been designed at Cambridge University and at the Los Alamos Laboratories.

Automatic Telephone Systems. The Bell Telephone Laboratories have contributed many fundamental ideas in the fields of digital computation, particularly in switching-circuit theory. Many of the functional operations basic to the telephone-communication problem are identical with digital-computer operations. In the telephone system a subscriber can address (call) one of many other addresses. To provide this service, the telephone system accumulates specific input information (dialed number), inspects and selects one of many possible routes of communication (trunks), switches to one of many possible memory locations (called number), and provides several additional logical choices: busy signals, ringing signals, unlisted numbers, and answering and hanging-up signals. Code translations, timing-memory scanning functions, and data accumulations and interpretations are other computerlike functions performed in a telephone system.

The DIAD telephone system has been built at the Bell Telephone Laboratories and demonstrates the use of modern electronic digital-computer techniques in a communications network.[39] Large rotating magnetic drums are used to store two types of data in this "computer."

1. Temporary information about calls being requested or in progress, such as the addresses of the requested and requesting telephones, and the trunk interconnections available for such calls

2. Permanent data: code identifications for all lines and trunks, code translations, programming techniques for placing and maintaining calls

These drums and associated elements constitute one of the two major blocks of the telephone system.[41] This drum block is the control section; it controls the operation of the second block, the switching network. This network is the maze of switching elements connecting all subscribers with a direct conversational path. The unusual feature of this electronic approach to telephone switching is that all the subscriber connections are made one at a time.[40] This technique reduces the amount of conventionally common control equipment and replaces it with high-speed electronic equipment. The corresponding switching array is enlarged to enable it automatically to connect subscribers and called lines. The specific paths for this connection are not

provided by the electronic controls; the control function is simply to indicate to the switching network the incoming and outgoing line addresses. An optimal method for designing these switching arrays has also been developed.[16]

General Applications. Electronic digital techniques are used for an extremely wide variety of special applications. To cite just a few, the television industry uses elegant digital counting techniques to generate the coded synchronizing timing sequences transmitted with the video signals. In the communications field, pulse-coded signals of many types are being employed. In transmitting video signals, for example, it may be practical to process pulse-code-modulated signals as computer data. One advantage of this approach is that the data may be relayed many times without deterioration.[51] The coding techniques developed to eliminate errors in computers and coding methods in general make it favorable to trade bandwidth for noise reductions.

Another area of great interest is that of print readers and pattern recognition. One of the basic approaches in many print-reading devices is to use a scanning system which produces a characteristic time sequence of pulses uniquely identifying each character of interest. Electronic digital smoothing, memory, and comparison techniques are then employed to identify the "read" characters and translate this information into a more compact digital form useful directly in data-processing equipment. Intelligent Machines Corporation of Arlington, Va., manufactures a variety of print-reading devices. Stanford Research Institute has constructed a machine to read standard-form characters on Bank of America Traveler's Cheques. A large general effort is being devoted to this effort because the manual-input phases in data processors are a bottleneck. Clark, Farley, Selfridge, and Dinneen at MIT have contributed a number of unique concepts in the field of recognizing time-sequence patterns.[14,55,19]

19.3. THE DIGITAL DIFFERENTIAL ANALYZER*

19.3.1. Introduction. The digital differential analyzer was originally developed in 1949 in order to provide a relatively inexpensive digital computer for the solution of differential equations. The first group of these computers were serial magnetic-drum devices which, for a large class of differential equations, operated in the time range of medium-speed general-purpose computers. Since that time, as a result of the intensive development and extensive proliferation of general-purpose computers and techniques, the development of digital differential analyzers has taken two directions: toward relatively small special-purpose devices for control and other related applications and, more recently, toward high-speed parallel devices for real-time simulations. A further development is the fusion of a digital differential analyzer and a general-purpose computer in order to provide for both types of computation within a single device.

The fundamental principles of operation of a digital differential analyzer may be viewed as synthesis of the principles of the mechanical analog differential analyzer developed by Dr. Vannevar Bush and the techniques of electronic digital computers. As in the Bush analyzer, solutions are generated by interconnecting—generally in a closed loop—elements that have a fixed mode of operation. These elements, which for the most part consist of integrators and servos, require a relatively small amount of equipment as compared with a general-purpose arithmetic unit. Furthermore, the interconnections serve to transfer changes in the values of the generated variables rather than the values of the variables themselves; hence the amount of information that must be processed is substantially reduced as compared with more orthodox digital techniques.

The most straightforward application of a digital differential analyzer, like that of a Bush analyzer, is to the solution of ordinary differential equations. However, the incremental nature of the intercommunicated information leads in a natural fashion to a generalized conception of incremental computation that is applicable to any computation involving continuous variables. Incremental computation thus considered is a numerical method coordinate with more traditional methods. Although the following material will emphasize the solution of differential equations, it should be noted that a digital differential analyzer can perform all the elementary arithmetic operations and can generate all the elementary functions and so has a wider field of application.

* By Max Palevsky.

The advantages and disadvantages of digital differential analyzers as compared with other types of computers stem to a large extent from the synthesis of digital and analog techniques. Thus digital differential analyzer computations are not limited in precision, as are analog computations, by component tolerances; rather, since all information within the computer is digital, precision is limited only by the capacity of the registers. On the other hand, because each of the computing elements of a digital differential analyzer has, like the parts of an analog computer, a fixed rather than a variable or programmed mode of operation, the flexibility of this type of computer is somewhat limited.

As compared with a general-purpose digital computer, a digital differential analyzer, like an analog computer, requires neither a close knowledge of numerical methods nor extensive programming. Furthermore, the digital differential analyzer solution is normally a close digital analog of the system under study so that the behavior of the various elements of the computer directly indicates the behavior of the system variables. More importantly, in many applications the frequency response, accuracy, and reliability requirements are such that the use of an analog or a general-purpose computer is either impractical or impossible. For example, certain real-time applications in the fields of aerodynamics, guidance, atomic physics, and mechanics require a greater accuracy than is obtainable with analog equipment while the complexity and frequency-response requirements could be met only by a general-purpose digital computer several orders faster than any that has thus far been contemplated. In other applications, particularly in the field of process control, the accuracy and reliability requirements preclude the use of analog equipment and severely restrict the number of components that may be employed. In many of these cases, a general-purpose digital computer that may be constructed from a given number of components is neither so powerful nor so flexible, within the solution times required by the application, as the comparable digital differential analyzer.

The material that follows is intended, primarily, as an exposition of the principles of operation and design of a digital differential analyzer. Some material on programming and applications is included; however, no attempt has been made at an extensive treatment of these subjects.

19.3.2. Principles of Operation. *The Integrator.* The basic element of a digital differential analyzer, the integrator, is essentially a fixed-program device employing a simple quadraturing scheme. One such scheme will be described before the range of applicable techniques is considered.

An integrator operates cyclically upon incremental inputs to produce incremental outputs. During any cycle, say the ith, the integrator receives a primary incremental input $(\Delta X)_i$, a secondary incremental input $(\Delta Y)_i$, and emits an incremental output $(\Delta Z)_i$ (see Fig. 19.1). The primary input $(\Delta X)_i$ indicates a change of a predetermined magnitude h_1 in the variable of integration X. If $(\Delta X)_i = +1$, X has increased by h_1; if $(\Delta X)_i = -1$, X has decreased by h_1 and with $(\Delta X)_i = 0$, X has not changed. Similarly, changes in the variable Y are indicated by $(\Delta Y)_i$ in units having the magnitude h_2. Unlike X, Y is allowed to change by more than a single increment during any given cycle. The magnitude of Y, however, is restricted to $-1 < Y < +1$.

Each integrator consists of two registers, a Y register and an R register. The Y register accumulates the value of Y, i.e., the initial value Y_0, plus the running total of the secondary inputs multiplied by h_2:

$$Y_i = Y_0 + h_2 \sum_{n=1}^{i} (\Delta Y)_n$$

Because Y may change by more than a single unit, the individual ΔY increments must be summed to form $(\Delta Y)_s$ before the addition that is to form Y_i. During the ith cycle, Y_i is added to the R register if $(\Delta X)_i = +1$ or subtracted from the R register

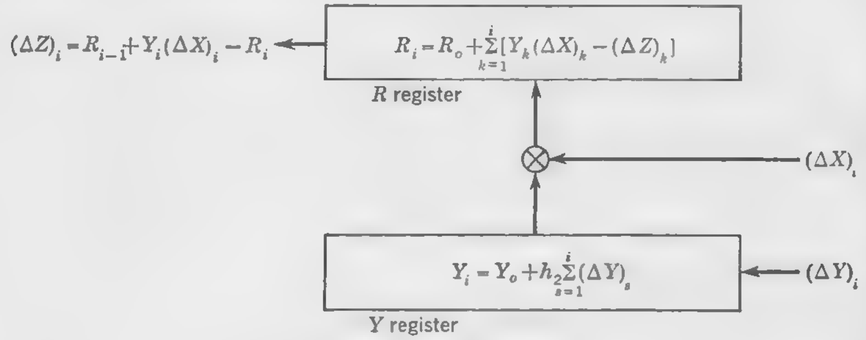


FIG. 19.1. An integrator.

if $(\Delta X)_i = -1$. If $(\Delta X)_i = 0$, R is unchanged. Given a sufficiently large register, the R value at the end of the ith cycle would be the sum

$$R_i = R_0 + \sum_{s=1}^{i} (\Delta X)_s Y_s \approx R_0 + \int_{X_1}^{X_i} Y\,dX$$

The R register, however, is restricted to the fractional portion of the sum while the integral portion, as shown in Fig. 19.2, takes the form of a series of incremental ΔZ outputs each having the predetermined magnitude h_3. Only a single increment can be generated during any cycle because of the restrictions on the values of R_i and Y_i. This increment may be either positive or negative depending upon whether the $R_{i-1} + Y_i(\Delta X)_i$ addition generates a positive or a negative carry into what is essentially the next most significant place in the variable. If neither condition holds, no increment is produced and the output is zero. Because the carry represents a subtraction of the incremental output, the remainder at the end of the ith cycle is

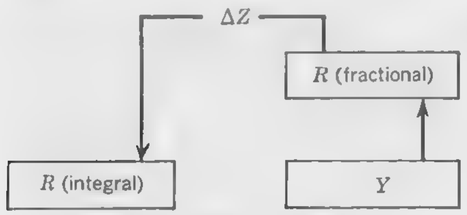


FIG. 19.2. The R register.

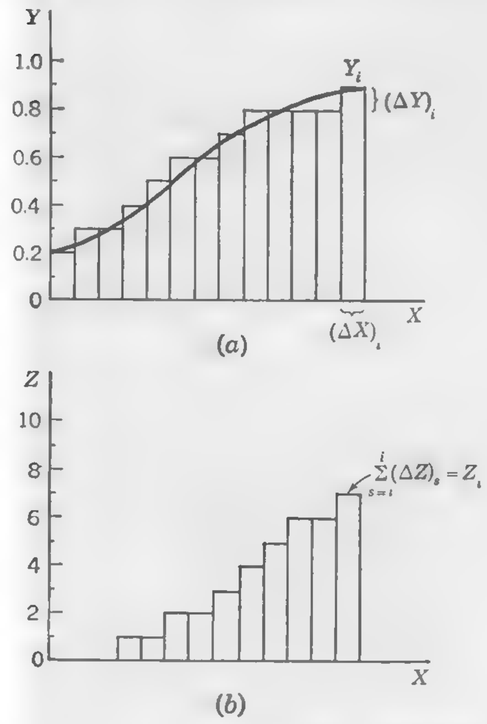
$$R_i = R_{i-1} + Y_i(\Delta X)_i - (\Delta Z)_i$$

The ΔZ increments, which represent the integral part of R, either serve as a secondary input to another integrator—in which case, as shown in the figure, R (integral) serves as the Y value of that integrator—or the ΔZ increments are used directly as a primary input elsewhere in the computer.

Although the integration scheme that has been outlined is not practical without certain modifications, it provides a simple conceptual model of the operation of a digital differential analyzer. Figure 19.3 is a graphical illustration of the operation of this scheme in which integration is represented as the summation of rectangular areas under a curve. Figure 19.3*a* shows the curve and the rectangular approximations while Fig. 19.3*b* represents the sum of these areas or the integral of the function Y. Figure 19.3*c* presents the same data in tabular form.

Examples of Integrator Interconnection. Before considering integrator operation in greater detail, it may be helpful to introduce several simple examples illustrating the use of integrators in the solution of differential equations. For schematic purposes, it is convenient to replace the finite increments, ΔX, ΔY, and ΔZ, by differentials

dX, dY, and dZ, respectively. Within any given problem these variables will be appropriately renamed. The schematic representation of an integrator is shown in Fig. 19.4. The relationship between the inputs and the output is $dZ = Y\,dX$. If



i	$Y_i(\Delta X)_i$	$(\Delta Z)_i$	R_i	Z_i
0	-----	---	0	0
1	0.2	0	0.2	0
2	0.3	0	0.5	0
3	0.3	0	0.8	0
4	0.4	+1	0.2	1
5	0.5	0	0.7	1
6	0.6	+1	0.3	2
7	0.6	0	0.9	2
8	0.7	+1	0.6	3
9	0.8	+1	0.4	4
10	0.8	+1	0.2	5
11	0.8	+1	0.0	6
12	0.8	0	0.8	6
13	0.9	+1	0.7	7

(c)

FIG. 19.3. Rectangular integration.

the output of the integrator is made the secondary input to another integrator, the Y register of the second integrator will accumulate the variable Z. The operation of the two integrators is such that $Z = \int_{X_0}^{X} Y\,dX + Z_0$ where Z_0 is the initial condition of Z and X_0 the initial condition of X. Figure 19.6 shows the more general case in which $Z = Z_0 + C_1(X - X_0) + \int_{X_0}^{X} Y\,dX$. Figure 19.7 illustrates the generation of e^X, i.e., the solution of $Y' - Y = 0$. Since the output of the integrator is $e^X\,dX = de^X$, the output provides the appropriate secondary input to the integrator. Figure 19.8 illustrates the integrator interconnections for the solution of $Y'' + Y = 0$, which generates the sine and cosine of X. The output of integrator 1 would normally be $\sin X\,dX = -d\cos X$. The minus sign at the point of the schematic of integrator 1, however, indicates that the sign of the output is reversed so as to yield $d\cos X$. With this as a secondary input, integrator 2 accumulates $\cos X$. The output of 2 is $\cos X\,dX$, which is exactly the required secondary input to integrator 1, i.e., $d\sin X$. Note that, if the output sign reversal is removed from integrator 1, the hyperbolic functions $\sinh X$ and $\cosh X$ would be generated. The next example (Fig. 19.9) illustrates the generation of $1/X$. With dX as the primary input and $1/X$ as the integrand, the first integrator generates $d\ln X$. With this as its primary input, $1/X$ as its integrand, and an output sign reversal, the second integrator generates $-1/X\,d\ln X = -1/X^2\,dX = d\,1/X$, which then becomes the secondary input to both integrators. As a final example, the multiplication of two variables U and V is shown in Fig. 19.10. The first integrator generates $U\,dV$, the second generates $V\,dU$, while the third integrator sums

FIG. 19.4. Schematic representation of an integrator.

$U\,dV + V\,dU = d\,(UV)$ and so holds

$$UV = \int_{U_0}^{U} V\,dU + \int_{V_0}^{V} U\,dV + U_0V_0$$

The Organization of a Serial Digital Differential Analyzer. A serial digital differential analyzer employs a single set of computing elements to process, in turn, a number of integrators. The device most commonly used to store the numbers associated with each integrator is a magnetic drum (or disk). The turning of the drum automatically sequences the information held on the magnetic surface. The number of integrators spaced around the drum within the limitations imposed by the diameter

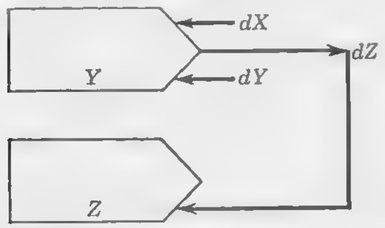


FIG. 19.5. The accumulation of dZ gives Z.

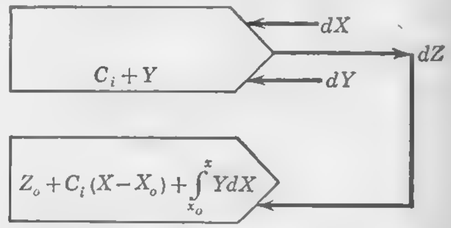


FIG. 19.6. General relationship of dZ and Z.

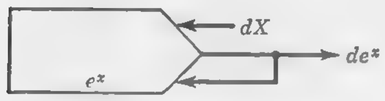


FIG. 19.7. Generation of e^x.

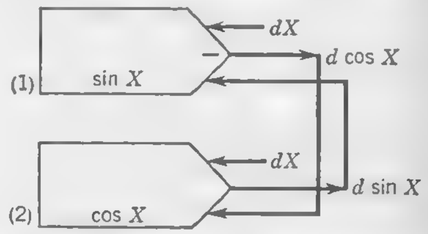


FIG. 19.8. Generation of sine and cosine.

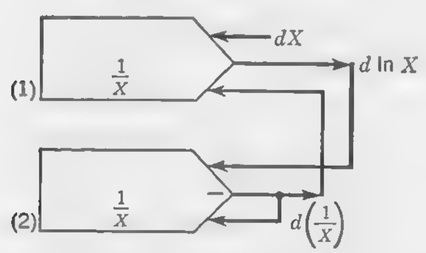


FIG. 19.9. Generation of $1/X$.

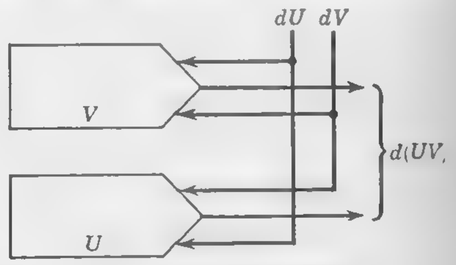


FIG. 19.10. Multiplication.

of the drum and the recording density is determined by the precision and iteration-rate requirements of the application. Typical values are 24 to 36 bits per integrator, 40 to 100 integrators, 60 to 100 iterations per second. A diagram of one possible configuration of the information stored on the drum is shown in Fig. 19.11 together with a detailed diagram of a single integrator. A recirculating memory is required for the arithmetic channels R and Y, since arithmetic information must at each pulse time be read from the drum and possibly modified before being returned to storage. Separate read and record heads are therefore necessary; furthermore, the remaining channels must also recirculate in order to remain in phase. A separate track with pulses evenly spaced around the drum synchronizes all operations so that the distance from the record to the read head in the direction of rotation determines the number of bits of delay. In some instances, the spacing of the heads presents mechanical difficulties that may be overcome by doubling the number of tracks and employing on each a single head which alternately reads and records. Two additional memory arrangements are often useful. The first, by making the distance between the recording and

reading heads some submultiple of the drum circumference, obviates the necessity for recirculating the nonarithmetic channels. Although only a fraction of the available storage is utilized, the limitation on the number of integrators is in some cases balanced by the resultant saving in circuitry and increase in speed. The second memory arrangement is especially useful in a computer for general computation since here the number of integrators required varies widely from problem to problem. Introducing a second set of read heads midway between the normal write and read heads permits the operation of the computer at twice the normal iteration rate with half the usual number of integrators. A switch on the control console permits the selection of either of the two modes of operation. This technique can be extended to any multiple of the operating speed at the expense of additional sets of read heads.

The Arithmetic Channels. The method of integration described on page 19–15 permits the R register to take on the values $-1.0 < R < +1.0$. The R value at any given time is essentially a remainder representing that part of the integral of $Y\,\Delta X$ that has not been transmitted as a ΔZ increment. Since the computations in an integrator affect the remainder of the computation only through these ΔZ increments, the R value represents the round-off error in Z. The magnitude of this error may be reduced by applying the rounding scheme normally employed in numerical calculations, i.e., generating an output whenever the absolute value of the R register equals or exceeds 0.5. The value of the R register and, consequently, of the round-off error is thus restricted to $-0.5 < R < +0.5$. However, the automatic subtraction of the output from the R register that is performed in the unrounded case by the carry must be replaced by an explicit subtraction whenever $+0.5 \leq R < +1.0$ or $-0.5 \geq R > -1.0$. The subtraction remains automatic if a carry is produced. In the following illustration of the two methods, the explicit subtractions are underlined.

Unrounded		Rounded	
	R_0 +0.00		R_0 +0.00
	Y_1 +0.76		Y_1 +0.76
$\Sigma\,\Delta Z = 0$	+0.76	$\Sigma\,\Delta Z = +1$ ←	+0.76
Error = +0.76		Error = −0.24	
	R_1 +0.76	+0.76 − 1.0 =	R_1 −0.24
	Y_2 +0.75		Y_2 +0.75
$\Sigma\,\Delta Z = +1$ ←	+0.51	$\Sigma\,\Delta Z = +2$ ←	+0.51
Error = +0.51		Error = −0.49	
	R_2 +0.51	+0.51 − 1.0 =	R_2 −0.49
	Y_3 −0.74		Y_3 −0.74
$\Sigma\,\Delta Z = +1$	−0.23	$\Sigma\,\Delta Z = +1$ ←	−0.23
Error = −0.23		Error = −0.23	
	R_3 −0.23		R_3 −0.23
	Y_4 −0.75		Y_4 −0.75
$\Sigma\,\Delta Z = +1$	−0.98	$\Sigma\,\Delta Z = 0$ ←	−0.98
Error = −0.98		Error = +0.02	
	R_4 −0.98	−0.98 + 1.0 =	R_4 +0.02
	Y_5 −0.74		Y_5 −0.74
$\Sigma\,\Delta Z = 0$ ←	−0.72	$\Sigma\,\Delta Z = -1$ ←	−0.72
Error = −0.72		Error = +0.28	
	R_5 −0.72	−0.72 + 1.0 =	R_5 +0.28
	Y_6 −0.73		Y_6 −0.73
$\Sigma\,\Delta Z = -1$ ←	−0.45	$\Sigma\,\Delta Z = -1$	−0.45
Error = −0.45		Error = −0.45	

The explicit subtraction in the case of $+0.5 \leq R < +1.0$ may be eliminated by increasing the number held in the R register by +0.5 so as to generate a carry whenever the true value of R, R_t is greater than or equal to 0.5. The carry also serves to signal an output of plus one. As a consequence, negative values of $R + Y$ indicate

that a negative increment must be generated and explicitly subtracted from R. At the start of any iteration, therefore, the R register is positive. The first column in the example below illustrates this modified arithmetic. The necessity for subtracting a plus one in the event that the output is negative may also be eliminated by employing the complement of Y if $Y\ \Delta X$ is negative. In this case, the sum of Y and R registers

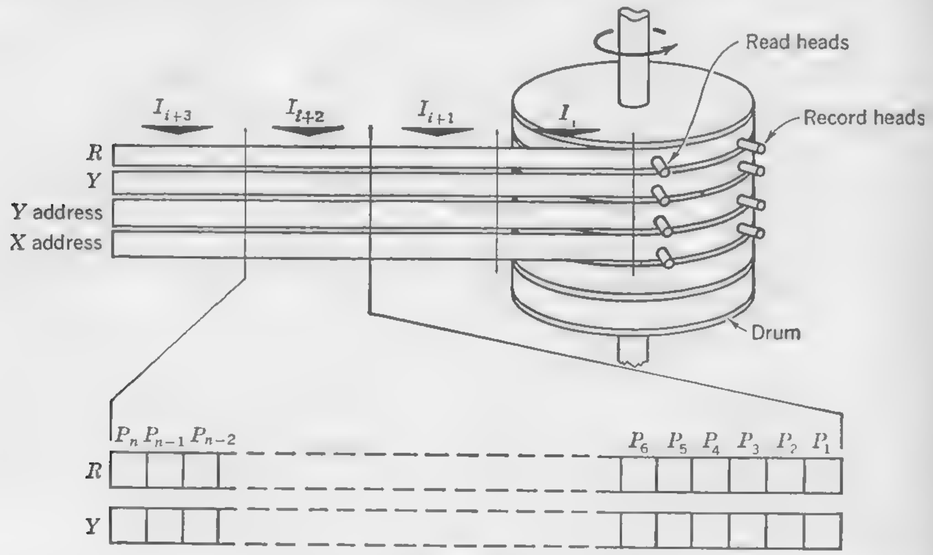


FIG. 19.11. Integrators on a drum.

is $(1 + Y_i) + (R_i + 0.5) = (R_i + Y_i) + 1.5$. An output is generated whenever this sum is less than 1, i.e., when $R_i + Y_i \leq -0.5$. Since no carry is generated in these cases, the plus one that was added to the sum by employing the complement of Y_i acts effectively as a correction for the negative output increment. This arithmetic is illustrated in the second column of the following example:

	R_0	+0.50		0.50
	Y_1	+0.76		+0.76
$\Sigma\ \Delta Z = +1$ ←		+0.26	+1 ←	0.26
Error = −0.24				
	R_1	+0.26		0.26
	Y_2	+0.75		+0.75
$\Sigma\ \Delta Z = +2$ ←		+0.01	+2 ←	0.01
Error = −0.49				
	R_2	+0.01		0.01
	Y_3	−0.74		−0.26
$\Sigma\ \Delta Z = +1$ ←		−0.73	+1 ←	0.27
Error = −0.23				
−0.73 + 1.0 =	R_3	+0.27		0.27
	Y_4	−0.75		−0.25
$\Sigma\ \Delta Z = 0$ ←		−0.48	0 ←	0.52
Error = +0.02				
−0.48 + 1.0 =	R_4	+0.52		0.52
	Y_5	−0.74		−0.26
$\Sigma\ \Delta Z = -1$ ←		−0.22	−1 ←	0.78
Error = +0.28				
−0.22 + 1.0 =	R_5	+0.78		0.78
	Y_6	−0.73		−0.27
$\Sigma\ \Delta Z = -1$		+0.05	−1	0.05
Error = −0.45				

The following is a résumé of the rules employed by the final scheme for determining the value of ΔZ:

1. If $Y\ \Delta X$ is positive,

$$(R_i + 0.5) + Y_i = (R_i + Y_i) + 0.5$$

A carry out of the most significant place in the addition of R and Y indicates that $(R_i + Y_i) \geq +0.5$; hence $\Delta Z = +1$. Lack of a carry indicates that $(R_i + Y_i) < +0.5$; hence $\Delta Z = 0$.

2. If $Y\ \Delta X$ is negative,

$$(R_i + 0.5) + (1 + Y_i) = (R_i + Y_i) + 1.5$$

Lack of a carry out of the most significant place in the addition of R and Y indicates that $(R_i + Y_i) \leq -0.5$; hence $\Delta Z = -1$. A carry indicates that $(R_i + Y_i) > -0.5$; hence $\Delta Z = 0$.

It should be noted that neither these rules nor the example above employs the sign of the number in the R register. The sign of R, therefore, need not be carried in the computation.

The implementation of the arithmetic that has been developed may now be considered. As the drum turns, the values of Y and R for successive integrators are read and modified before being rerecorded. As Y is read, it is summed algebraically

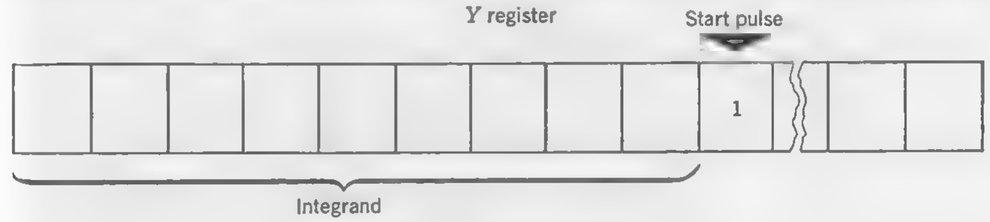


FIG. 19.12. Start pulse.

with ΔY, and $Y + \Delta Y$ is recorded in the Y channel. Negative values of Y and ΔY are represented as complementary numbers; so that the sum, if negative, is also a complement.

The position at which the Y increments are added is determined by the relative magnitude of Y and the ΔY increment (i.e., by the precision of computation as expressed in the scaling of the problem). This position will vary from integrator to integrator in most problems and so must be specified in some manner. One means of establishing this point is to place a 1—called a "start pulse"—directly to the right of the position to be specified, leaving the remainder of the integrator empty as shown in Fig. 19.12. If the number-representation system is such that each digit has at least one nonzero bit (e.g., excess-three binary-coded decimal) the least significant digit of Y is easily identified without this additional pulse. The precision of the integrand is not diminished by this rounding operation since it is limited by the significance of the ΔY increments. In certain cases, to be discussed on page 19–46, the accuracy of the integral of the function is affected by the bias entailed in rounding off. These effects may be virtually eliminated by holding either the start pulse or a number which indicates the pulse time of the least significant digit of Y in another channel. This number in conjunction with a counter can be employed to generate a signal which initiates the $Y + \Delta Y$ addition. The register that holds the sum of the ΔY inputs during this addition is generally shorter than the Y register. The ΔY register may be artificially extended with a minimal amount of equipment by employing the sign digit as the addend in the positions to the left of the sign of ΔY provided plus is represented by zero and minus by $P - 1$, where P is the radix. The addition of $Y = -0.86347$ and $\Delta Y = -7 \times 10^{-5}$, for example, has the form

$$\begin{array}{l} 9.13653 \\ 9.99993 \\ \hline 9.13646 = -0.86354 \end{array}$$

Having formed $Y + \Delta Y$, the result must be added to R if $\Delta X \neq 0$. In the event that ΔX is negative, the $P - 1$ complement of $Y + Y$ is used and the carry into the least significant digit of R is set. The fractional portion of the sum is recorded in the R channel while the integral portion in conjunction with the signs of Y and ΔX determines the sign and existence of the ΔZ increment. It is fairly simple, at this point, to provide for an output sign reversal under the control of a bit stored somewhere within the integrator.

The operation of the computer is simplified if overflows due to improper scaling are sensed and faulty arithmetic operations are detected. In either case, computation should be halted and the fault indicated by a light on the control console. An integrator may intentionally be programmed to overflow in order to halt computation at a specified value of some variable. Any of the well-known types of error detection may be employed; however, because of the increase in equipment entailed, only the simplest are generally used. In a four-bit binary-coded decimal computer, for example, the occurrence of any of the six unused binary codes is easily detected and does not require any change in the arithmetic of the computer.

The Output Distribution System. As each integrator is processed, the resultant output must be held so as to be available as an input to subsequent integrators. If static storage elements are employed, n integrators require $2n$ binary storage elements since each output may take on three values. Not only is this number of static storage

P_{12}	P_{11}	P_{10}	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1
I_i	I_{i-1}	I_{i-2}	I_{i-3}	I_{i+4}	I_{i+3}	I_{i+2}	I_{i+1}	t			

FIG. 19.13. Information on the Z line.

P_{12}	P_{11}	P_{10}	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1
1	8	7	6	5	4	3	2	t			

FIG. 19.14. Z line.

elements excessive for an n of almost any magnitude, but a complex switching system is required for both reading and recording the stored increments. An alternate approach is to employ a short track on the drum for storing the outputs as they occur. The operation of this track may be understood by considering a computer in which the number of integrators equals the number of pulse times within each integrator. A track one integrator in length that precesses one bit each integrator time provides, in this case, for storing each new output as it is produced and dropping each old output as it is replaced. The positioning of outputs within the track is relative to the integrator time at which the track is read. In what follows only a single track (Z line) will be considered on which a 1 signifies the existence of an output. An identical track holds the sign of the increment.

The Z line briefly described in the previous paragraph is generally not feasible both because the number of bits per integrator usually does not equal the number of integrators and because the Z line must provide, in addition to integrator outputs, the computer independent variable ("computer time" or simply "time") and, in some cases, its negation ("minus time"). The latter difficulty may be obviated by replacing a specified integrator output by computer time; no loss in capacity results since at least one integrator in every problem is used merely to collect some quantity which either is to be printed and serves no further function, or is to control the initiation of input or output (see Art. 19.3.5). The solution to the former difficulty depends upon the relative number of integrators and pulse times within each integrator. The first case to be considered is that in which the number of pulse times is greater. For example, with 12 pulse times and 8 integrators, one possible arrangement of information on the Z line in integrator I_i is shown in Fig. 19.13. Times P_7 to P_5 are empty, P_4 holds computer time (a plus 1), and P_{12} holds the output of the integrator I_i itself.

		Z_I record										Z_{II} read
I_1	1	–										
	2	–										
	3	–										
	4	t										
	5	2										
	6	3	2									
	7	4	3	2								
	8	5	4	3	2							
	9	6	5	4	3	2						
	10	7	6	5	4	3	2					
	11	8	7	6	5	4	3	2				
	12	1	8	7	6	5	4	3	2			
I_2	1		1	8	7	6	5	4	3	2		
	2			1	8	7	6	5	4	3	2	
	3				1	8	7	6	5	4	3	2 t
	4	t				1	8	7	6	5	4	3 2
	5	3					1	8	7	6	5	4 3
	6	4	3					1	8	7	6	5 4
	7	5		3					1	8	7	6 5
	8	6			3					1	8	7 6
	9	7				3					1	8 7
	10	8					3					1 8
	11	1						3				1
	12	2							3			–
I_3	1	–	2	1	8	7	6	5	4	3	t	–
	2	–									3	–
	3	–										3 t
	4	t										3
	5	4										4
	6	5										5
	7	6	5									6
	8	7		5								7
	9	8			5							8
	10	1				5						1
	11	2					5					2
	12	3						5				
I_4	1	–							5			
	2	–								5		
	3	–									5	t
	4	t										5 4
	5											5

Direction of time ↓ (Drum rotation →)

Z line configuration of $P_4 I_2$ ←

FIG. 19.15. Operation of the Z line.

The pulse in P_4, the independent variable of the computer, supplies the output to which all other outputs are traceable much as the motion of the main shaft in a mechanical differential analyzer supplies the motion to which all other motions can be traced. Placing the output of I_i in P_{12} is convenient since the output of an integrator is available only at the end of the integrator time and so may be placed directly onto the Z line rather than stored in some static device. Assuming the following arrangement of information in integrator 1, the information in $P_{5/11}$ comes from the drum, P_{12} holds the output of the integrator and P_4 is set to 1 in this and all other

integrators. Figure 19.15 illustrates the operation of the Z line. Eleven pulse times after being recorded from Z_I the information is read by Z_{II} during integrator 2 time. At P_4 the information being read—the old output of integrator 2—is replaced by t and at P_{12} the new output of integrator 2 is recorded. As the figure indicates, the process is cyclical so that, after eight integrators, eight precessions have occurred and the cycle is restarted.

The 11 pulse times of delay may be supplied partly by delays in the reading and recording circuitry and partly by the switching circuitry. Generally, an 11-pulse track is not practical; so that a delay of $12k + 11$ pulse times must be used, where k is the smallest integer which yields a viable system. In this case, information recorded during integrator I_i does not arrive at the read head until I_{i+k+1}. $k + 1$ bits of information, therefore, are "old," only one of which is replaced by the output of integrator I_{i+k+1} itself. The remaining outputs, if they are to be universally available, must be held statically in order to be available during integrator times I_{i+1} through I_{i+k}. Furthermore, they must be placed on the Z line during these integrator times; otherwise the output of each integrator will appear only in integrators which fall in the same residue class modulo $k + 1$.

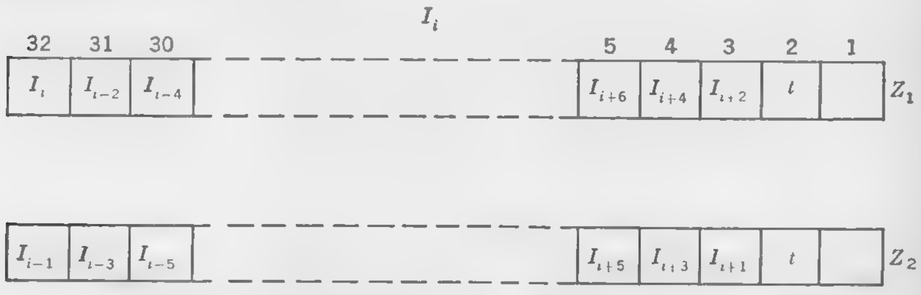


FIG. 19.16. Z line with more integrators than pulse times.

Thus far, the ordering of information on the Z line has proceeded under the assumption that each integrator produces an output (1 or 0) during each memory cycle. If computation is interrupted, however, new outputs are not produced; so that after one cycle the information contained on the Z line would be destroyed. This may be remedied in several ways, the two most straightforward of which are to stop precession and to replace artificially the new output of the integrator by the old. The former of these involves adding an extra bit of delay to the Z line whenever computation is halted and connecting the record circuitry directly to the read circuitry. When computation is resumed, both these changes must be removed at the same point in the computation at which they were entered. The latter remedy takes advantage of the fact that the previous output of an integrator is read from the Z line during the integrator time in which the new value would normally be computed. In the present example, as may be seen from Fig. 19.15, the old value is read during P_4. This value may then be remembered and substituted during idling for the normal output of the integrator.

A further complication is added to the Z line in the event that the number of integrators is larger than the number of pulse times. If the former is 60 and the latter 32, one possible arrangement of information is shown in Fig. 19.16. The position of any output within any integrator is given by Fig. 19.16; it may also be computed using the formula

$$P = (q - I + 30) \bmod 30 + 2 \tag{19.6}$$

where q is the desired integrator, I the integrator in which it is to be found, and P the pulse time. The use of (19.6) requires several other rules:

1. p is rounded up.
2. $30 \bmod 30 = 30$.
3. If I is even, even q's occur in Z_1, odd q's occur in Z_2, and vice versa if I is odd.

In implementing this scheme, it is advantageous to use a single track on the drum

with one recording head and two reading heads. Ignoring the physical problem of minimum allowable head spacing, one possible arrangement of information on the Z line at P_{32} of any given integrator is shown in Fig. 19.17. The recirculated information is taken from the Z_1 reading head; so the position of any given integrator will be shifted by one pulse time each successive time it appears in either of the read heads. Since this occurs in every other integrator, the given integrator output will precess thirty times in each line or sixty times each cycle. As in the previous case, I_i is placed on the line at the final pulse time P_{32}, and a 1 is recorded each P_2 time to provide computer time. In the event that physical considerations require longer Z lines, essentially the same system can be extended.

In both the examples that have been considered, empty spaces remain on the Z lines. The employment of these in connection with incremental inputs will be discussed on page 19–55. Finally, it should be noted that physically the Z lines may often be placed on the drum between the reading and recording heads of the long lines.

The Address Unit. The incremental inputs to a given integrator must be collected and, in the case of secondary inputs, summed before the integrator is to be processed. If the addresses are placed in long lines, the portion of the drum that is read during integrator I_i must contain the addresses for integrator I_{i+1}. A 1 in any pulse time indicates that the corresponding content of the Z line is to serve as an input. With multiple Z lines, universal addressing requires multiple address lines. As the secondary inputs are picked up, they are sent to a bidirectional counter which, in effect,

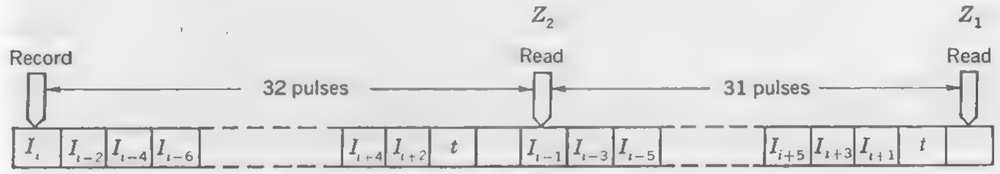


FIG. 19.17. Z line with two reading stations.

sums them and represents negative numbers as complements. If multiple address lines are employed, the counter must provide for multiple counts during a single pulse time. After a certain point it may be more efficient to extend the counting process over several integrator times so as to reduce the maximum input to the counter during any one pulse time, even though this requires the use of several counters. In either case, at the end of I, the secondary input to integrator I_{i+1} has been summed by the counter. This sum, however, cannot be employed until the start pulse occurs, whereas the counter must (after being cleared) start collecting the secondary inputs to the following integrator. A register is therefore required to hold the summed secondary input until it is to be used.

Thus far only the secondary input has been considered. The primary input differs in that it is limited to the output of a single integrator (or to computer time). If the primary input is the sum of the outputs of several integrators, an additional integrator (see page 19–39) is required to produce the sum. If the particular Z line which holds the desired integrator is specified, the addressing of the primary input requires only one additional line. Furthermore, the required specification may be held by some portion of the line which corresponds to empty Z-line positions. An alternate scheme for addressing the primary input is possible in the event that enough unused bits are available in the secondary-input address lines for specifying the number of any of the integrators. This number is placed in integrator I_{i-2}. When read, a portion of the number sets a counter which counts down during integrator I_{i-1}. The zero configuration of the counter in conjunction with the remainder of the number specifies the Z-line location that is to be employed as the primary input to I_i. A memory line, therefore, is eliminated at the cost of a counter and its associated circuitry.

The Control Unit. The control unit provides synchronizing and control signals. In addition to the clock, which is read from the drum, other synchronizing signals are required in order to identify certain pulse times within each integrator, e.g., the

pulse time in which the output of the integrator is formed and placed on the Z line. Furthermore, at least one integrator time must be identified in order that computation may be started and stopped at the same point. The monitoring of computation may also require that any of the remaining integrators be selectable. Either counters or drum channels may be used to develop these synchronizing signals, the choice being a function of the available circuitry and the exact requirements. In a small special-purpose digital differential analyzer a single synchronizing track is probably sufficient, whereas in a large computer for more general applications, counters are generally more efficient.

Once computation is started, little or no control of the process is required since the program is effectively built in. It may be necessary, however, to stop computation, to vary initial conditions automatically (see page 19–40), or perhaps to effect a rescaling of the entire problem (Art. 19.3.8). In any of these cases, computation must be restarted automatically. In addition, either an overflow of one of the integrands or the detection of some functional error may be employed to stop the computer. Here, however, the restarting of computation requires manual intervention.

Organization of a Parallel Digital Differential Analyzer. The integrators of a parallel digital differential analyzer, unlike those of a serial computer, are individual units which operate simultaneously. In this respect they are analogous to the operational amplifiers of an analog differential analyzer. One advantage of parallel operation—the higher iteration rate—is obvious. Not so obvious is the resultant increase in flexibility and simplification of maintenance. A serial digital differential analyzer has a fixed number of integrators which cannot be increased except by intercoupling two computers. Aside from the cost of such a procedure, the synchronizing of two drums raises serious difficulties. Furthermore, modifying the operation of selected integrators, which may be required by certain problems, is difficult in a serial digital differential analyzer whereas a parallel device requires only the addition of modified types of parallel elements. Maintenance is simplified since faulty units are easily replaced and repaired without rendering the entire computer inoperative. The disadvantage of a parallel computer is the increase in the number of computing elements, which involves both an increase in cost and a decrease in reliability.

The increase in cost of a parallel as compared with a serial digital differential analyzer cannot be justified, in general, without increasing the iteration rate of the parallel computer beyond that which results directly from the parallel mode of operation: A parallel computer with an iteration rate of 6,000 integrations per second does not, except in special applications, warrant the manifold increase in cost as compared with a serial computer of comparable capacity with an iteration rate of 100 integrations per second. A further increase of from 10 to 25 is required which, in terms of present-day techniques, implies a pulse-repetition rate of the order of 2 Mc for a parallel computer with serial arithmetic. The difficulties usually associated with frequencies of this order are to a high degree obviated, however, because each integrator can be constructed as a compact unit and only low-frequency signals need pass between units.

Arithmetic Unit. The arithmetic operations which must be performed by a parallel digital differential analyzer are essentially the same as those of a serial computer. As in the serial machine, addition within each integrator is serial, not only because parallel addition is prohibitively costly, but because serial addition lends itself to some form of inexpensive storage, such as electrical delay lines, as opposed to unit-by-bit devices.

Output Distribution System and Address Unit. In a parallel computer the outputs that result from any iteration are used immediately and so need not be stored. The output distribution system and the address unit can therefore be combined into a single unit which may employ a plugboard as its central element. At the end of each iteration, consequently, the inputs to an integrator arrive in parallel. The first few pulse times of the following iteration may be used to sum the secondary inputs in order to avoid the parallel addition of more than three inputs in a single pulse time. Since the number of integrators which require multiple secondary inputs is, in any given problem, a fraction of the total number, it is wasteful to supply an adder in

conjunction with each integrator. Rather, this adder may be provided either as a separate unit which is attached as needed, or as an integral part of only some of the integrators.

The Control Unit. Because of the high pulse-repetition rate, it is generally not feasible to distribute signals from a central control in order to identify pulse times within the integrator word. One synchronizing signal, however, is necessary for identifying the beginning of the iteration. It and the clock are sent to each integrator and employed to develop the remaining timing signals. Similarly, the starting and stopping of computation may present difficulties if controlled from a central location. These difficulties may be overcome by gating the one synchronizing signal and allowing it to control the starting and stopping of computation. An alternate scheme, which also avoids the necessity for distributing another time-critical signal, is to generate relatively long starting and stopping signals which are then transformed within each integrator into the exact signals which control computation.

Arithmetic of a Binary-transfer Digital Differential Analyzer. Both the parallel and the serial digital differential analyzer considered thus far produce ΔZ increments which may take on the values $+1$, 0, and -1. A less specialized scheme, called variously "binary transfer," "binary intercommunication," and "binary output," employs only two values, $+1$ and -1. Each integrator therefore produces a nonzero output during each iteration, the sign of the output being determined by the sign of the sum of Y and R. The R-register algorithm is

$$R_i = R_{i-1} + Y_i \Delta X_i - \Delta Z_i \tag{19.7}$$

as in the ternary case; however, unlike the ternary case, the range of the round-off error $-1 \leq \epsilon \leq +1$ cannot be reduced without introducing an additional output state. The following example illustrates the application of binary-transfer arithmetic to the example employed on pages 19–15 and 19–19. In every iteration but the third, the

	R_0	$+0.00$
	Y_1	$+0.76$
$\Sigma \Delta Z = +1$ ←		$+0.76$
Errors $= -0.24$		
$+0.76 - 1.0 =$	R_1	-0.24
	Y_2	$+0.75$
$\Sigma \Delta Z = +2$ ←		$+0.51$
Error $= -0.49$		
$+0.51 - 1.0 =$	R_2	-0.49
	Y_3	-0.74
$\Sigma \Delta Z = +1$ ←		-0.23
Error $= -0.23$		
	R_3	-0.23
	Y_4	-0.75
$\Sigma \Delta Z = 0$ ←		-0.98
Error $= +0.02$		
$-0.98 + 1.0 =$	R_4	$+0.02$
	Y_5	-0.74
$\Sigma \Delta Z = -1$ ←		-0.72
Error $= +0.28$		
$-0.72 + 1.0 =$	R_5	$+0.28$
	Y_5	-0.73
$\Sigma \Delta Z = -2$ ←		-0.45
Error $= +0.55$		

subtraction of ΔZ entails an explicit arithmetic operation. The adder required to perform this operation may, as in the ternary case, be eliminated (see below). On the other hand, the adder can be used in conjunction with an additional register—the S register—to provide each integrator with a constant multiplier by weighing the

value of ΔZ that is subtracted from R. The algorithm for the R register in this case is

$$R_i = R_{i-1} + Y_i\,\Delta X_i - S\,\Delta Z_i$$

so that

$$\sum_{p=1}^{i} \Delta Z_p = \frac{1}{S}\left(R_0 - R_i + \sum_{p=1}^{i} Y_p\,\Delta X_p\right)$$

where $(1/S)(R_0 - R_i)$ represents the round-off error. Because the number held in the S register must have the same range of values as the numbers in R, viz., $-1 < S < +1$, $1/S$ is greater than 1 in absolute value. As a consequence, the value of S may be employed in lieu of a start pulse to effect the proper valuation of secondary-input increments. That is, the secondary inputs to each integrator may be added to the last place at the right end of the Y register and this incorrect valuation compensated for by adjusting S. For example, if the inputs each represent 10^{-3} unit relative to the value of the Y register but are entered at the 10^{-7} place, multiplying the constant to be held in S by 10^{-4} will effectively multiply the inputs (and the output) by 10^4. Because the number in S may be employed both for scaling and for multiplication by a constant, it is generally called a "scaling multiplier." It should be noted that, if S is negative, the sign of the output of an integrator is effectively reversed. Other advantages of multiplication by numbers greater than 1, particularly as related to a decimal computer, will be discussed on page 19–29.

The disadvantage of employing scaling multipliers is that the unmultiplied value of the integrator output is not available. The output of every integrator, except that holding the lowest derivative, must, in general, serve as an unmultiplied input to the next lower order integration and, in addition, be multiplied by a constant in order to form an input to the integrator that is to close the loop. For this reason, constant multiplication should, except in special applications, be separate from the integration process. It is interesting to note that scaling multipliers are applicable only in a binary-transfer computer since only here is the output dependent upon the sign and not the value of R. If more than two output states are admitted, the value of R at which outputs are generated becomes a function of S.

The implementation of scaling multiplication, as was noted above, requires an adder and a register and eliminates the start-pulse circuitry. Although this adder may increase the additions that must be performed during each pulse time from two to three, the over-all increase in equipment is smaller than for any of the other constant multiplication schemes to be discussed on page 19–29. The additions are generally performed in the following order:

$$Y_{i-1} + \Delta Y_i$$
$$(Y_{i-1} + \Delta Y_i)X_i - S\,\Delta Z_{i-1}$$
$$[(Y_{i-1} + \Delta Y_i)X_i - S\,\Delta Z_{i-1}] + R_{i-1}$$

The ΔZ employed must, of course, be the last output of the integrator rather than, as in Eq. (19.7), the ΔZ about to be produced. For this reason, the first iteration may increase the round off by the value in S unless R is started at the value $S\,\Delta Z_0$, where ΔZ_0 represents the state of the Z line before the start of computation. An alternative is to block the addition of S during the first iteration.

As noted above, if scaling multiplication is not introduced, the explicit subtraction of the ΔZ in Eq. (19.7) may be eliminated. The first step to this end is to replace the true values of Y and R, Y_t and R_t, by $1 + Y_t$ and $1 + R_t$. The unit's digit in both cases effectively represents and replaces the sign. The $R + Y$ addition therefore has the form $(1 + Y_t)\,\Delta X + (1 + R_t)$. If ΔX is positive, this reduces to $2 + Y_t + R_t$, which produces a carry out of the addition in the most significant place whenever $Y_t + R_t$ is positive. The carry not only indicates an output of $+1$ but effectively subtracts $+2$ (a $+1$ to compensate for the $+\Delta Z$ and $+1$ to compensate for the 1 added to Y_t). On the other hand, if $Y_t + R_t$ is negative, no carry results, indicating an output of -1. The requisite subtraction of -1 is taken care of by the $+1$ added to Y_t. If ΔX is negative, a further step is necessary, viz., the complementation of

$(1 + Y_t)$. The $R + Y$ sum in this case has the form

$$2 - (1 + Y_t) + (1 + R_t) = 2 + (-Y_t + R_t)$$

and the positive ΔX analysis applies. The following example illustrates the application of this arithmetic to the example above.

	R_0	1.00
	Y_1	1.76
$\Sigma\,\Delta Z = +1$		0.76
Error $= -0.24$		
	R_1	0.76
	Y_2	1.75
$\Sigma\,\Delta Z = +2$		0.51
Error $= -0.49$		
	R_2	0.51
	Y_3	0.26
$\Sigma\,\Delta Z = +1$		0.77
Error $= -0.23$		
	R_3	0.77
	Y_4	0.25
$\Sigma\,\Delta Z = 0$		1.02
Error $= +0.02$		
	R_4	1.02
	Y_5	0.26
$\Sigma\,\Delta Z = -1$		1.28
Error $= +0.28$		
	R_5	1.28
	Y_6	0.27
$\Sigma\,\Delta Z = -2$		1.55
Error $= +0.55$		

19.3.3. Generation of Special Functions. *Multiplication of a Variable by a Constant.* In addition to integration, a digital differential analyzer must often perform other kinds of operations. These include the multiplication of a variable by a constant or by another variable, the generation of inverse functions, and the limiting of functions between specified values. In a serial computer, provisions for performing each of these special operations are included in the common arithmetic unit and selected by proper coding. In a parallel computer, special computing elements may be supplied to perform these functions. The terms "integrator" and "integrand" thus take on extended meanings which encompass more than the simple process of integration.

FIG. 19.18. Multiplication by a constant.

There are, in general, five schemes for the multiplication of a variable by a constant in a digital differential analyzer in addition to the scaling multiplier described on page 19–27. In the most straightforward of these, the constant is placed in an integrator, the variable provides the primary input, and there is no secondary input (Fig. 19.18). Although this is perfectly adequate, it is somewhat wasteful of computer capacity, particularly as compared with analog differential analyzers in which constant multiplication is performed by potentiometers. The four remaining schemes provide more efficient alternatives.

The first of these (Fig. 19.19) employs a Y and an R register as in a normal integrator but eliminates the circuitry required for supplying Δy inputs and for performing the $Y + \Delta Y$ addition. In a serial computer, each such multiplier is associated with a

specific integrator so that the primary input to the multiplier is the output of the integrator without the mediation of any output-distribution device (Z line). Only the output of the multiplier need be provided for general distribution while the unmultiplied output is usually available as an input to only the succeeding one or several integrators. If no constant is entered, the multiplier is automatically +1.

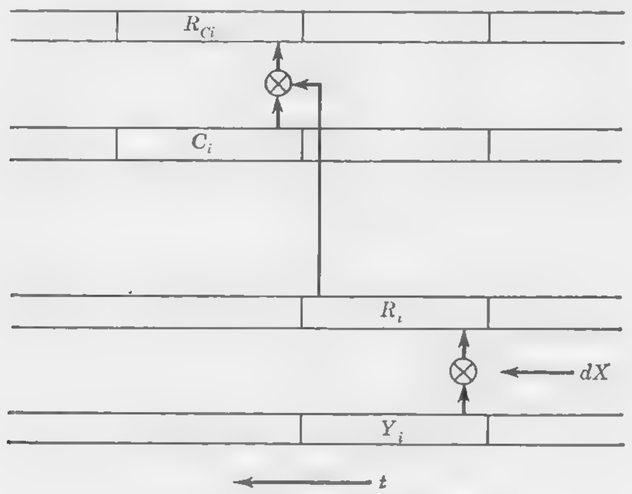


FIG. 19.19. Constant multiplication.

If this scheme is employed in a parallel computer, the individual multiplying units resemble integrators except that no provisions are made for the processing of secondary inputs or for the transfer of information to the read-out medium. Furthermore, in both the parallel and serial cases the initial conditions remain unchanged and so need not be reset if a problem is to be rerun.

A second alternative scheme of constant multiplication takes advantage of the fact that the integrands in certain problems require only a portion of the integrator word length; the remainder may then be employed to hold a constant. An integrator of this type, called a "split integrator," is illustrated in Fig. 19.20*a*. The point at which the split occurs is generally fixed so that only a single bit—somewhere on the address lines—is required to signify this type of operation. Furthermore, no additional arithmetic circuitry is required. Only the control circuits must be modified so as to process information as shown in Fig. 19.20*b*. Additional flexibility may be provided by storing the unmultiplied output so as to make it available during the next integrator time.

The third alternate is employed primarily to mitigate the scaling inefficiencies of a decimal radix. In a binary computer, the number of scaling factors available between one and million is approximately 20 since $2^{20} \approx 10^6$. In a decimal computer the number available is six. If the numbers to be scaled are assumed to have a rectangular probability distribution, the efficiency E of a radix r is

$$E = \frac{K_1}{r-1}\int_1^r \frac{X}{r}\,dX + \frac{K_2}{r^2-r}\int_r^{r^2} \frac{X}{r^2}\,dX + \frac{K_3}{r^3-r^2}\int_{r^2}^{r^3} \frac{X}{r^3}\,dX + \cdots$$

where the K's are weighting factors. Integrating,

$$E = \frac{K_1}{2}\left(1+\frac{1}{r}\right) + \frac{K_2}{2}\left(1+\frac{1}{r}\right) + \frac{K_3}{2}\left(1+\frac{1}{r}\right) + \cdots$$

Because each term is identical and the sum of the weighting factors is 1,

$$E = \tfrac{1}{2}\left(1+\frac{1}{r}\right)$$

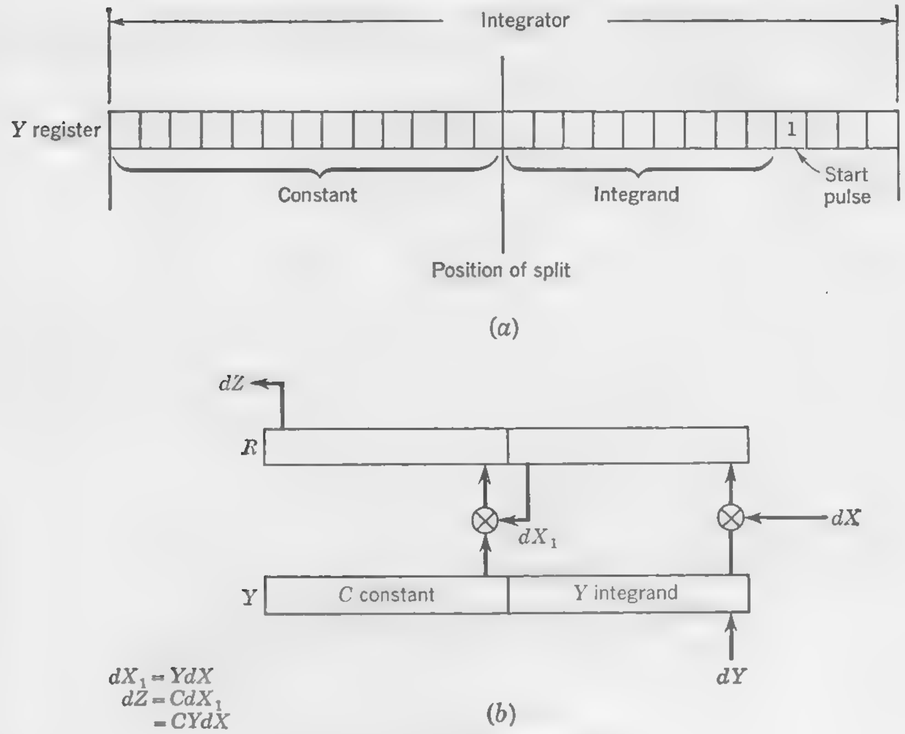


FIG. 19.20. Split integrator.

As might be expected, E is the average of the extreme efficiencies. With other probability distributions this function changes. For example, if the logarithms of numbers are assumed to have a rectangular probability distribution,

$$E = \frac{1-(1/r)}{\ln r}$$

The efficiency is a measure of the solution time required for a given precision or the precision resulting from a given solution time. What is of importance, however, is not this absolute efficiency, but rather the comparative efficiency of various practical radices. Using the first probability assumption, $r = 2$ yields $E = 0.75$, while $r = 10$ yields $E = 0.55$, so that the relative efficiency is 1.36. Although the increase in efficiency of the binary radix is small for a single integrator, in interconnecting integrators this efficiency is often multiplied, with the result that the relative efficiency for a given problem may become very large. For example, the efficiency of the interconnection in Fig. 19.21 is the product of the efficiencies of the individual integrators. A generalized estimate of the relative over-all efficiency of the two radices is difficult to make; in individual cases, however, the analysis is straightforward.

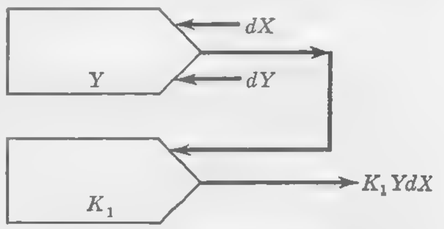


FIG. 19.21. Scaling multiplication.

The effects of the inefficiency of a decimal radix may be mitigated by the addition of output multipliers which, unlike the other constant multipliers considered in this section, *increase* the output rate. In order to be useful in a decimal computer, these multipliers must be provided in pairs which have 10 as a product. Considerations of both conceptual simplicity and circuit complexity limit such pairs to 2 and 5 or 2½

and 4. Furthermore, the additional efficiency resulting from providing the second pair after one pair has been adopted is too small to warrant its adoption. The efficiency of the 2, 5 pair is 0.733; hence

$$\frac{E(r = 2)}{E[R = 10(2, 5)]} = 1.023$$

and

$$\frac{E[R = 10(2, 5)]}{E(r = 10)} = 1.333$$

The scaling for a single problem in the three systems will be found in Art. 19.3.7.

In addition to increasing scaling efficiency, an output multiplier which increases the output rate of an integrator effectively decreases the round-off error by that factor. This follows from the decreased range of the remainder in the R register. A further advantage of this scheme is that it requires no additional memory capacity. Rather, it operates by changing the capacity of the R register. Thus, if the multiplier is 5, the capacity of the R register is contracted to $0 \leq R < 0.2$; with a multiplier of 2 to $0 \leq R < 0.5$; with a multiplier of 4 to $0 \leq R \leq 0.25$; and with a multiplier of 2.5 to $0 \leq R < 0.4$. The advantage of the 2, 5 scheme is obvious—fewer digit positions must be altered. The size of the register may be changed in one of two ways, by altering the machine representation of zero and detecting overflows in the normal fashion or by altering the overflow point without changing the zero representation. In either case, provisions must be made for initially placing the correct "half-full" value in the R register in order to effect the proper round-off.

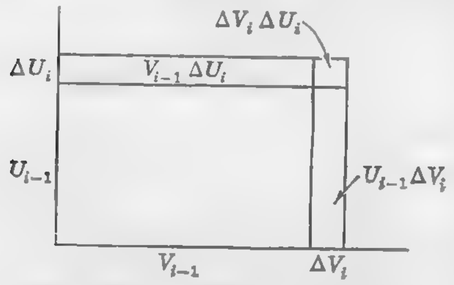


FIG. 19.22. Variable multiplication.

The fourth alternative output-multiplier scheme is appropriate only in special-purpose digital differential analyzers of a certain type. In such applications, a given rate must be multiplied by a variety of constants. Rather than employing the hitherto described multipliers, less equipment is required and higher speeds are possible with the so-called "operational" multiplier.[25] This requires a single counter for the variable plus a register for each of the constants. Some attempts have been made to employ this scheme for integration by allowing the value in the register to vary; however, the process generates too large an error to be practical.

Variable Multiplication. Increments of the product of the two variables U and V may be calculated in a digital differential analyzer by forming

$$\Delta(UV) = U \Delta V + V \Delta U$$

using the integrator interconnections shown in Fig. 19.10. If the inputs ΔU and ΔV are generated in a serial computer by integrators which are processed before the multiplying integrators, during the ith cycle

$$\Delta(U_i V_i) = (V_{i-1} + \Delta V_i)U_i + (U_{i-1} + \Delta U_i)\,\Delta V_i$$
$$= V_{i-1}\,\Delta U_i + U_{i-1}\,\Delta V_i + 2\,\Delta V_i\,\Delta U_i$$

Graphically, this means that the small square in Fig. 19.22 is counted twice. Negative values of ΔV_i and ΔU_i yield similar results, while if either ΔU or ΔV is zero, the result is exact. The error in the product may be avoided by spacing the multiplying integrators between the integrators which generate ΔU and ΔV (Fig. 19.23). Here,

$$\Delta(U_i V_i) = (V_{i-2} + \Delta V_{i-1})\,\Delta U_i + (U_{i-1} + \Delta U_i)\,\Delta V_i$$
$$= V_{i-1}\,\Delta U_i + U_{i-1}\,\Delta V_i + \Delta U_i\,\Delta V_i \quad (19.8)$$

That this is the correct product may be verified by referring to Fig. 19.22.

An alternate method of mechanizing Eq. (19.8) involves adding an additional register Y_2 to the normal integrator configuration and providing for the operations shown in

Fig. 19.24. The ΔV input is added to Y_1 and the result is added into R under the control of ΔV, as in the normal integrator. This forms the $U_{i-1}\,\Delta V_i$ and $\Delta U_i\,\Delta V_i$ terms of Eq. (19.8). The ΔV input is also added into Y_2, but before this sum is formed, the value of Y_2 is added into R under the control of ΔU. This forms the $V_{i-1}\,\Delta V_i$ term of Eq. (19.8). The output of this modified integrator therefore represents increments of UV so that, as opposed to the scheme of Fig. 19.23, no device for adding the outputs of two integrators is required if $\Delta(UV)$ is to be employed elsewhere as a primary input. Additional advantages include a reduction in the number of incremental signals that must be handled by the output-distribution system and a reduction from four to three in the number of registers and the number of adders required in a parallel computer for multiplication. This latter advantage is partially canceled by the complication that results from performing three rather than two serial additions in each pulse time. In a serial computer it is difficult to justify the additional adder and memory capacity except in special applications that require a relatively large number of multiplications. A more important disadvantage in the three-register scheme is the scaling restriction that it entails. Unlike the normal scheme, in which the range of the variables to be multiplied has no special restriction, the three-register scheme requires, if double outputs are to be avoided, that the absolute value of the sum of Y_1 and Y_2 not exceed 1. The resultant scaling, as compared with the normal scheme, will in many cases require at least twice as much computation time. When not employed for multiplication, the additional register permits a single integrator to

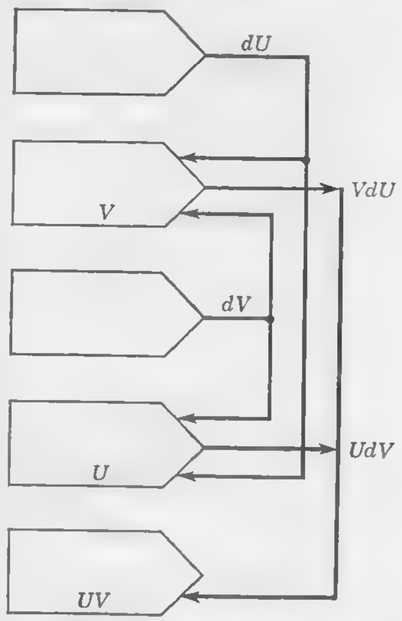


FIG. 19.23. Generation of products.

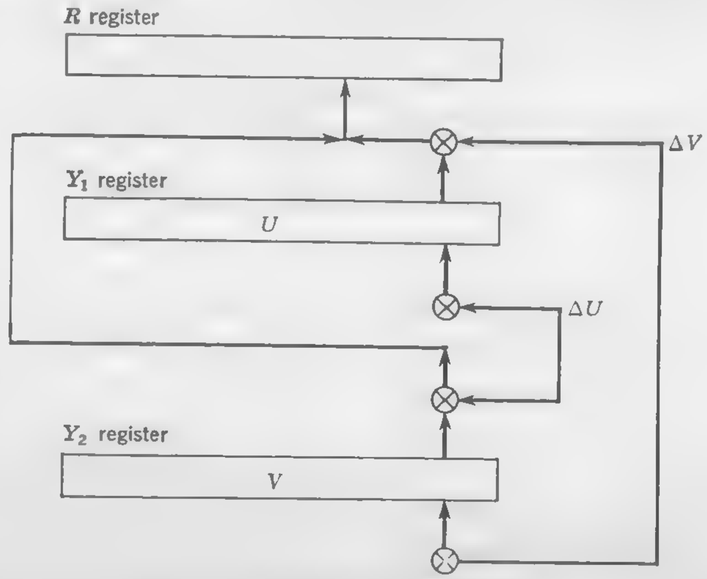


FIG. 19.24. Alternate multiplication.

form $K_1\,\Delta U + K_2\,\Delta V$, where K_1 and K_2 are constants and U and V are variables, or $(Y_{i-1} + \Delta Y_i + Y_{i-1})\,\Delta X_i = 2(Y_{i-1} + \frac{1}{2}\,\Delta Y_i)\,\Delta X_i$. The latter is the algorithm for the interpolative form of the trapezoidal rule; additional equipment is required in

order to generate the extrapolative form. In some applications the extrapolative form of integration need not be included, e.g., a control computer which requires no closed loops since the functions to be integrated are supplied to the computer rather than generated by it.

In a serial computer, the first exact multiplication scheme may be inconvenient; while in a parallel computer, because all integrators are processed simultaneously, it cannot be realized by programming alone. The second scheme, on the other hand,

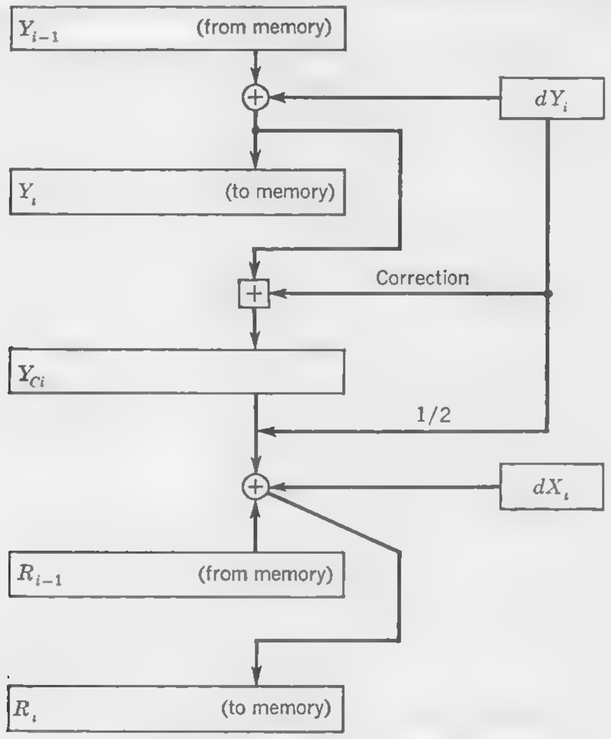


FIG. 19.25. Multiplication for parallel systems.

involves economic and scaling difficulties. A third approach forms the exact product by using two modified integrators each of which form

$$(Y_{i-1} + \tfrac{1}{2}\,\Delta Y_i)\,\Delta X_i$$

so that
$$\Delta(U_iV_i) = (U_{i-1} + \tfrac{1}{2}\,\Delta U_i)\,\Delta V_i + (V_{i-1} + \tfrac{1}{2}\,\Delta V_i)\,\Delta U_i$$
$$= U_{i-1}\,\Delta V_i + V_{i-1}\,\Delta U_i + \Delta U_i\,\Delta V_i$$

Because the two inputs ΔU and ΔV result from the same iteration, the method is applicable to parallel computers and simplifies programming in serial computers. The implementation of this method of multiplication necessitates the generation of $(Y_c)_i = Y_{i-1} + \tfrac{1}{2}\,\Delta Y_i$ as well as the normal $Y_i = Y_{i-1} + \Delta Y_i$. Once $(Y_c)_i$ is added into R, however, it need not be remembered; so that no additional storage is required. $(Y_c)_i$ may be developed directly from the algorithm for $(Y_c)_i$ or indirectly by forming $Y_i - \tfrac{1}{2}\,\Delta Y_i$. A third alternative employs the fact that ΔY_i is either $+1$, -1, or 0; so that $\tfrac{1}{2}\,\Delta Y_i$ is $+\tfrac{1}{2}$, $-\tfrac{1}{2}$, or 0. The addition of $(Y_c)_i$ into R when $(Y)_i \neq 0$, therefore, has a $\tfrac{1}{2}$ as the least significant digit of the addend. If this $\tfrac{1}{2}$ is merely appended to the Y_i value in order to form $(Y_c)_i$, the result is exact whenever $\Delta Y_i = -1$ since

$$Y_i + \tfrac{1}{2} = Y_{i-1} + \Delta Y_i + \tfrac{1}{2} = Y_{i-1} + \tfrac{1}{2}\,\Delta Y_i = (Y_c)_i$$

If $Y_i = +1$, however,

$$Y_i + \tfrac{1}{2} = Y_{i-1} + \Delta Y_i + \tfrac{1}{2} = Y_{i-1} + \tfrac{1}{2}\,\Delta Y_i + 1 = (Y_c)_i + 1$$

Subtracting 1 in this case produces an exact result.

Figure 19.25 illustrates one implementation of the scheme. Y_{i-1} and Y_i are summed, with the result going back to the memory and to a correction adder. This adder requires little circuitry since it must perform only two operations—addition of 0 and -1. The resultant $(Y_c)_i$ has $\tfrac{1}{2}$ appended to it and the addition into R takes place in the normal fashion.

Servos. In addition to integrators, the generation of certain functions in a digital differential analyzer requires the use of some type of digital servo. In generating U as a function of V, for example, where U is defined implicitly as a function of V by $F(U, V) = 0$, a servo must be employed. The use of an integrator as a proportional servo for the solution of such problems is shown in Fig. 19.26. If $\delta F(U, V)/\delta U$ is different from zero and opposite in sign from dV, incremental changes in V that result in an $F(U, V)$ that is different from zero will generate incremental changes in U until $F(U, V)$ is again equal to zero. The rate at which increments of U are generated is proportional to the error signal held in integrator 1. The summation of these increments yields the value of U.

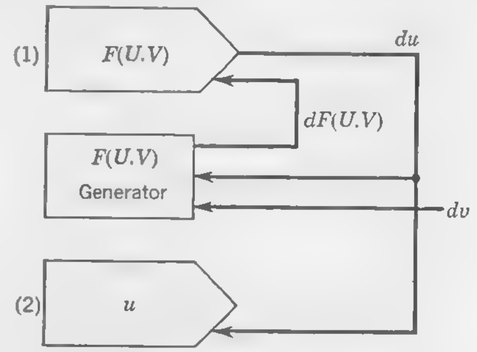


FIG. 19.26. Proportional servo.

Because the response of a proportional servo of this kind is generally too slow, most digital differential analyzers employ a scheme which produces a considerably faster response. In this scheme, the servo produces a maximum output rate whenever the integrand is different from zero, regardless of the magnitude of the integrand. The conditions upon $F(U, V)$ remain the same as does the diagram in Fig. 19.26, except that the integrator is replaced by a servo. Figure 19.27 contains two examples of the use of servos.

A servo effectively has no primary input. If a servo is modified so that a primary input is employed to control the output, the result is a decision integrator. In connection with decision integrators it is usually convenient to introduce a form of limiting which may be called "clipping." The point at which clipping occurs is some predetermined value of the integrand of the decision integrator. A convenient method for specifying this value is to add an additional bit—a unit's digit—to the left of the radix point so that the range of the integrand becomes

$$+2 > Y \geq -2$$

The output $(\Delta Z)_i$ of the decision integrator during the ith cycle with a primary input $(\Delta X)_i$ is then

0	if	$+2 > Y_i \geq +1$
$(\Delta X)_i$	if	$+1 > Y_i > 0$
0	if	$Y_i = 0$
$-(\Delta X)_i$	if	$0 > Y_i > -1$
0	if	$-1 \geq Y_i \geq -2$

Figures 19.28 to 19.30 illustrate various uses of decision integrators.

In a computer employing ternary transfer, the operation of servos and decision integrators may be implemented by examining the integrand for a nonzero digit and, if one exists, producing an output with the sign of the integrand in the servo case, with the sign of the product of the signs of the integrand and the primary input in the decision-integrator case. The circuitry in the two cases is identical if in the former the primary input is automatically set at $+1$.

In a serial computer, one set of servo and decision elements is time-shared; so that by placing a 1 in some predetermined spot—usually one of the unused address-line positions—any integrator may be made to operate in either fashion. The $Y + R$

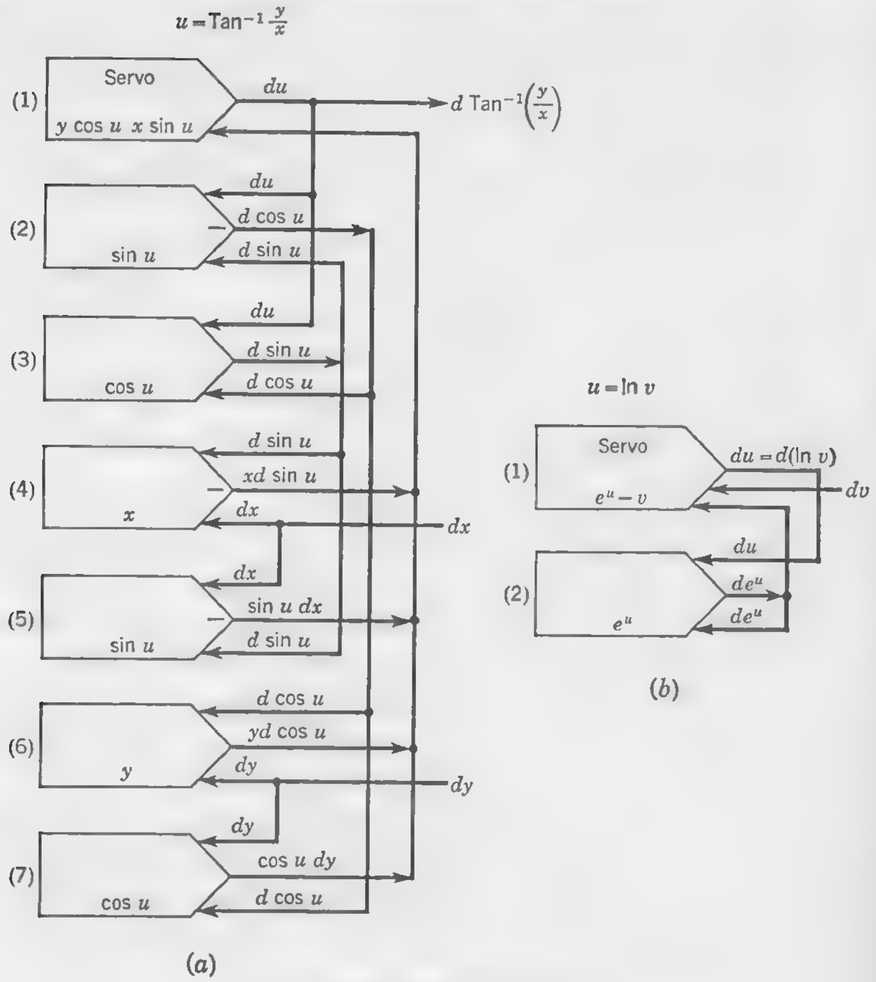


FIG. 19.27. The use of servos.

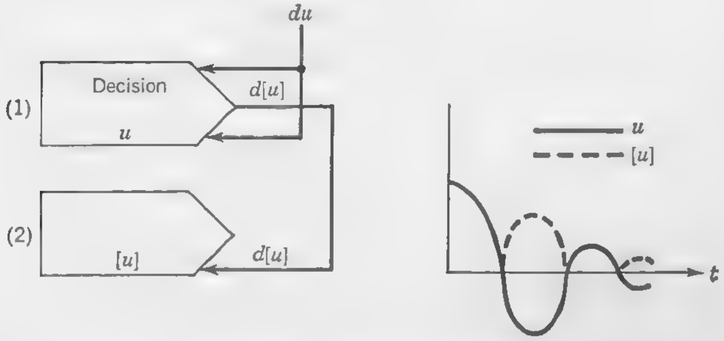


FIG. 19.28. Decision integrator.

adder continues to operate, but the information in the R register is ignored. Because this adder and register are superfluous, separate servo-decision units are appropriate in a parallel computer. The clipping action described above is provided by examining the unit's digit in conjunction with the sign in order to determine if the combination of the two represents a 1. In the event that it does, the existence portion of the output is set to zero.

The integrators of a binary-transfer computer may be operated as servos by taking advantage of the number system described on page 19–27. In that number system, the largest positive number, $1 - e$, is one increment away from the largest negative number, -1. In a decimal computer the former is represented by

$$1.999 \ldots 9$$

the latter by

$$0.000 \ldots 0$$

If the Y register of an integrator that is employed as a servo is started at -1, the output of the integrator will be a series of -1's which will eventually cause some

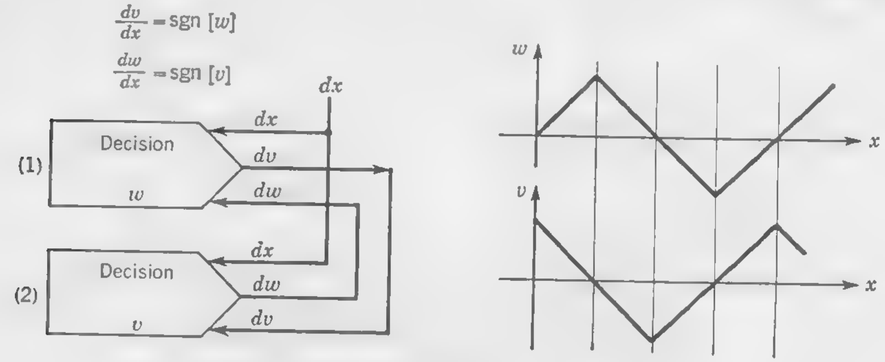


FIG. 19.29. Use of decision integrators.

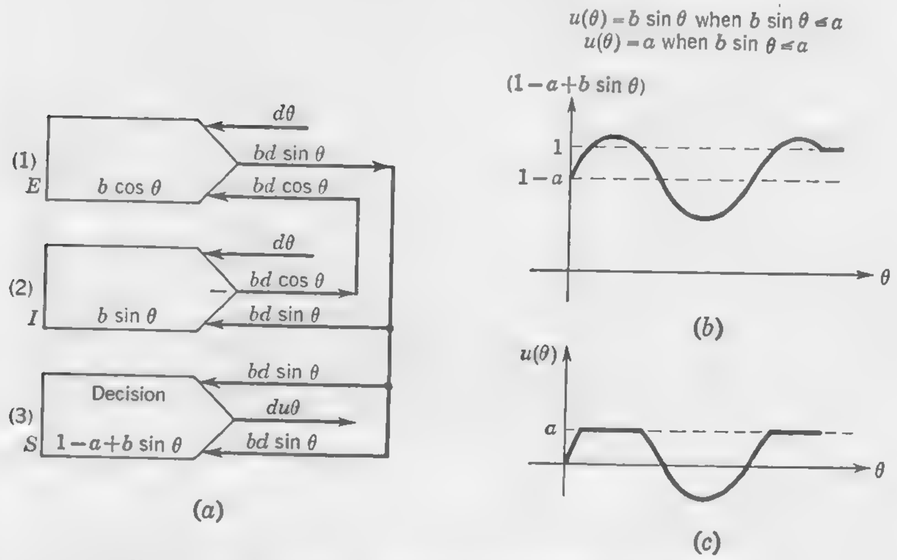


FIG. 19.30. Change of equations with decision integrator.

quantity to be subtracted from the integrand, driving it to a large positive value. A series of $+1$'s will result, again driving the integrand toward a negative value. The output is therefore either a maximum positive or a maximum negative rate.

Several problems arise with a servo of this type in addition to those produced by the "hunting" which results from the inability of the system to represent a zero error signal. The first is a lack of sensitivity in the output of the servo to changes in the error signal near the null position. Since the input to a servo is for the most part a zero rate, the integrand will vary on successive cycles by K pulses, K being a function of the relative configurations of the secondary inputs. This variation will take place around some mean value M such that $M + K/2$ is a large positive number and $M - K/2$ is a large negative number. As long as the mean value lies within $K/2$

increments of -1, the output rate will be zero. If, however, the mean value moves so that it is more than $K/2$ pulses away from -1, the output of the servo becomes either a maximum negative or a maximum positive rate. As these outputs operate through the servo loop, they eventually drive the integrand back to the zero output rate range. Since an input must drive the integrand out of the zero rate range before it is effective, the result is a certain "slack" or "looseness" in the servo. No comparable problem exists in a ternary-transfer computer. A further difficulty in a binary-transfer computer results from the Z-line configuration at the start of a computation. This configuration—generally either all $+1$'s or all -1's—is predetermined in order to ensure computational repeatability. The secondary inputs of a servo are therefore generally erroneous during the first cycle of computation and may produce several outputs before returning to the zero-output region. In effect, thereafter, the two functions which are being servoed are not kept equal but differ by a small fixed amount. If only the derivatives of the functions are of interest, this may not be objectionable. If, however, the functions themselves are being computed, an error results. This error may sometimes be reduced by running the problem for a few cycles, noting the behavior of the servos, and restarting the problem with slightly different initial values in the servos.

The error which results because the mean value of the integrand is not -1 (0.000 . . . 0 in the computer language) has been ignored thus far because it is generally too small to affect more than a few of the digits at the extreme right-hand end of the R register. If, however, the servo is operated as a decision integrator, the mean value of the integrand may be such that the error in the register results in an erroneous output. For example, if $Y_0 = 1.90$ and $R_0 = 1.00$, 11 iterations will produce a series of ten $+1$'s followed by an erroneous -1. Because the error increases with the magnitude of the integrand, it is not feasible to employ servos of this type as limiters. The functions performed by decision integrators in a ternary computer must therefore be supplied in a different fashion in a binary-transfer computer.

One approach to the limiting function is to employ an additional register to hold the limiting value. If the integrand is less than the number in this register, the output is identical to the secondary input; in the event that the integrand is larger, the output is zero. By holding the secondary input in the primary-input register, setting the primary input to -1, and adding Y to the limiting value rather than to R, the limiting function may be implemented with a minimal amount of additional equipment. The negative primary input results in the addition of the complement of Y to the limiting value so that the sign of the sum may be employed to control the output of the limiter. In the event that this sum is negative, the requisite zero rate may be generated by using the carry resulting from successive additions of 1 into the most significant digit of R. In a digital differential analyzer which stores the initial conditions as well as the values of all variables (see page 19–40) the initial-condition register may be employed to hold the limiting value. In this event, however, the initial condition of the variable must be entered separately rather than from the initial-condition register.

Limiting in this manner is particularly applicable to a digital differential analyzer with binary transfer since the number system permits a single integrator to restrict functions between any two limits. For example, if a sine wave of amplitude 1 is to be clipped at $+0.57$ and -0.83, the limiting integrator may be started at -0.17 and the limiting value set at 0.40. Whenever the amplitude exceeds 0.57, the integrand will exceed 0.40 and a zero output will result. When, on the other hand, the amplitude is less than -0.83, the integrand will attempt to go more negative than -1.0 and a large positive integrand will result. The output in this region, consequently, will also be zero. The disadvantage of this type of limiter is that it can limit a variable only as a function of the value of the variable itself and not, like the clipper limiter of a ternary-transfer digital differential analyzer, as a function of the value of another variable. To provide for this latter case, it is necessary to introduce another special function.

One form that this special function may take is the signum function which permits the sign of the output of an integrator to be reversed if the integrand of the immediately preceding integrator is negative. Using this function, absolute values may be generated in an alternate manner to that shown in Fig. 19.28. When U becomes negative, the output of (2) is reversed, provided the integrators are processed serially as shown in Fig. 19.31. In a parallel computer, an error is produced each time the sign of U changes. The signum function is also very convenient for switching integrators into and out of computation in problems in which different differential equations are employed depending upon the region of the solution.

In addition to limiting a function by clipping, another form of limiting, which is analogous to the action of a stop in a physical system, is

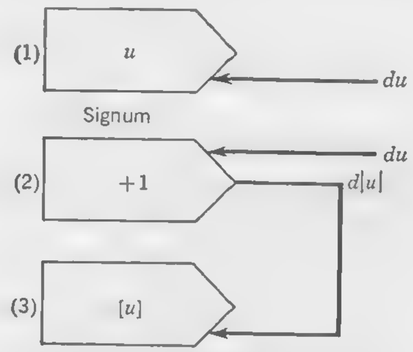


FIG. 19.31. The signum function.

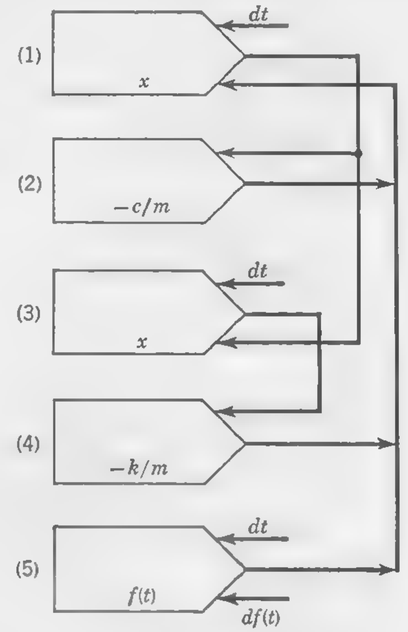


FIG. 19.32. A second-order equation.

sometimes required. For example, the system represented by the equation

$$m\ddot{X} + c\dot{X} + kX = f(t) \tag{19.9}$$

may have a stop at $X = S$. The integrator diagram for Eq. (19.9) is shown in Fig. 19.32. When the system reaches the point $X = S$, X must be set to zero and computation resumed. However, $\dot{X}$ must not be allowed to change until the sign of the sum of the inputs to (1) in any iteration is opposite to the sign of the value of $\dot{X}$ before it was reset. The resetting operation will then be discussed on page 19–40. The blocking of the inputs may be provided in a ternary-transfer computer by holding the sign of $\dot{X}$ statically and setting another bit of static storage until the stored sign and the sign

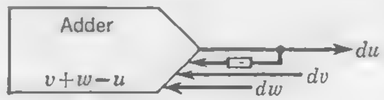


FIG. 19.33. Addition.

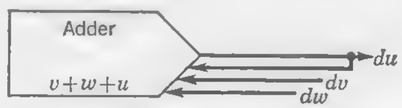


FIG. 19.34. Subtraction.

of $\dot{X}$ differ. At this point normal operation resumes. In a binary-transfer computer some form of filtering is required in order to distinguish between a zero rate and a true change in the sign of $\dot{X}$.

Adders. An adder, or summing integrator, is required to sum either the primary or the secondary inputs to an integrator whenever the number of such inputs exceeds the capacity of the input registers or counters. Essentially, an adder is a servo which employs its own output as a secondary input. If provisions are made in a ternary computer for automatically negating this output before it is used as an input, the summing integrator operates as shown in Fig. 19.33, where $dU = dV + dW$. An input to the adder causes an output of the same sign that is subtracted on the next cycle from the number held in the V register. Thus, if three positive increments are

entered, three positive outputs will result on successive cycles and the adder will be returned to zero. If the sign change on the input is not provided, the adder will operate to form the negation of the sum with the aid of a sign reversal, so that $dU = -(dV + dW)$.

In a binary-transfer computer, since a positive input to a servo produces a negative output, the two comparable schemes are as shown in Fig. 19.35. In both instances special summing-integrator circuits may be avoided by developing the negation of the sum. In another sense, too, adders are not required since they may be eliminated completely by providing sufficiently large input registers for both primary and secondary inputs. In a parallel computer, this would be especially desirable in order to avoid the delay introduced by the addition process. However, except in certain special-purpose computers, the extra equipment involved is too costly.

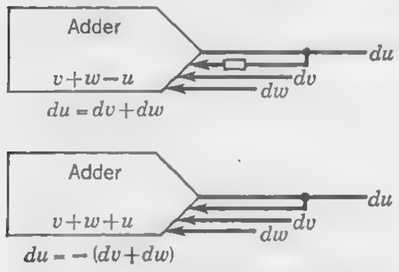


FIG. 19.35. Addition and subtraction.

Storage and Variation of Initial Conditions. It is often convenient in a digital differential analyzer to store the initial conditions of all variables in addition to the integrands proper. The register used for this purpose is generally called the I register. The inclusion of such a register not only facilitates the initial check-out of a problem but provides a quick and efficient means of producing the successive solutions of a family of the problems since only the new parameter values need be entered at the start of each solution. In a parallel computer with a high iteration rate the solution time of certain problems may be short enough to allow the solution to be viewed on an oscilloscope provided the problem can be continually recycled. Here, then, not only manual resetting but some form of automatic resetting of initial conditions is required.

The automatic resetting of initial conditions is also required if a digital differential analyzer is to be capable of automatically solving the class of problems which includes split-boundary and double-integral equations. In addition, these problems require facilities for incrementing the initial conditions as a function of an ongoing solution. By restricting such increments to the output of a single integrator, an addressing system similar to that employed for primary inputs will suffice. One bit of this addressing system must be employed to indicate which integrators are to be reset since, unlike the case in which recycling occurs, only selected integrators are reset. Finally, an adder is needed to form the sum $I + \Delta I$. The operation of the system is illustrated by the following two examples:

1. The equation $$\ddot{y} = At + e^y + C$$

is given together with the following initial conditions:

$$\dot{y}_0 = -B$$
$$t_0 = 0$$

The problem is to find y_0 such that at some time t_f

$$\dot{y}(t_f) = y(t_f) = 0$$

The solution proceeds as shown in Fig. 19.36*a* by making a trial estimate of y_0, ${}_1y_0$, and generating the corresponding curve until $\dot{y} = 0$. The final value of y, ${}_1y_f$ is then employed to form

$${}_2y_0 = {}_1y_0 - m({}_1y_f)$$

where m is a damping coefficient such that y_f does not diverge. Successive initial values of y are generated until the terminal conditions are met.

The integrator interconnection employed to solve the problem is shown in Fig. 19.36*b*. The contents of the initial-condition register are indicated at the top of each

integrator schematic. Integrators 1, 2, and 3 are used to generate the changes which will produce the next y_0. Integrators 4, 5, 6, 7, and 8 are used to generate $y(t)$ for each successive value of y_0. Integrator 9 controls the point at which the initial conditions are reset. As each set of initial conditions is placed into the computation channels, the new estimate of y_0 is entered into the Y register of integrator 1 where

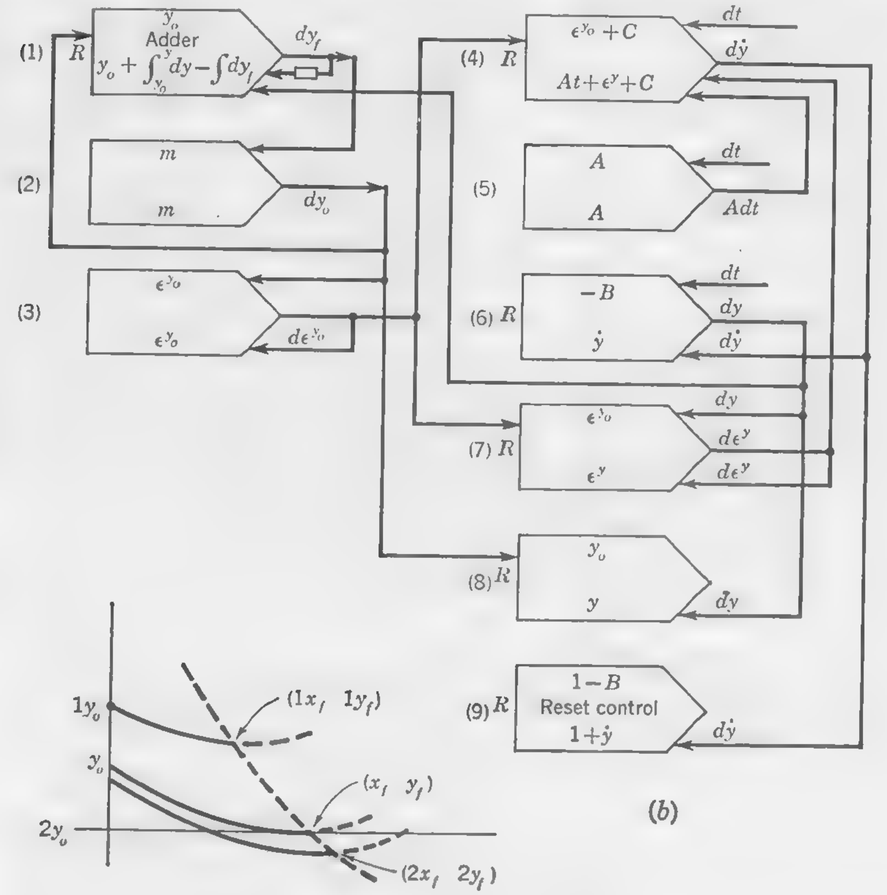


FIG. 19.36. A nonlinear example.

it is augmented by the incremental changes in y that result from the computation. At the end of the ith trial, therefore, the total incremental output of integrator 1 is

$${}_iy_0 + \int_{{}_iy_0}^{{}_iy_f} dy$$

which is exactly the magnitude of the error in y. It should be noted that only those integrators are reset which have an R on the left-hand side of their schematics.

2. In order to evaluate the double integral

$$V = \int_{3\frac{1}{2}}^{\pi} \int_{3\frac{1}{2}}^{\pi/2y} \cos(xy)\, dx\, dy$$

the volume of a number of slabs (see Fig. 19.37) must be summed. Each slab has a thickness Y and an area on one side equal to

$$\int_{1_0}^{\pi/2y_n} \cos(y_n x)\, dx$$

If there are m slabs,

$$y_1 = \tfrac{1}{2}$$
$$y_2 = \tfrac{1}{2} + y$$
$$y_n = \tfrac{1}{2} + (n-1)y$$
$$y_m = \tfrac{1}{2} + (m-1)y = \pi$$

The initial value of cos (xy) for each slab is cos $(y/2)$ which is generated by integrators 1, 2, and 3 of Fig. 19.38. The adder generates the incremental changes of y from y_{n-1} to y_n while the corresponding changes in $y/2$, cos $y/2$, and sin $y/2$ are sent to the initial-condition registers of integrators 4, 5, 6, 7, and 9. The sum of the volumes of the successive slabs is formed in integrator 8, computation on each slab being halted whenever the integrand of 9, cos $(xy) - 1$, equals -1. This corresponds to the limit of integration $X = \pi/2y_n$. The entire integration process is advanced by the incremental outputs of integrator 10 until cos $(y/2) - 1$ equals -1. At this point—where $y = \pi$—computation is halted by the limiting action of the decision integrator.

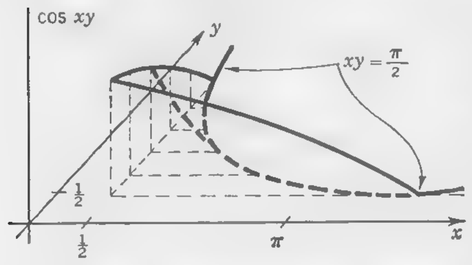


Fig. 19.37. Double integration example.

19.3.4. Error. *Truncation Error.* The computational errors generated in a digital differential analyzer are a combination of the single-step errors generated in each

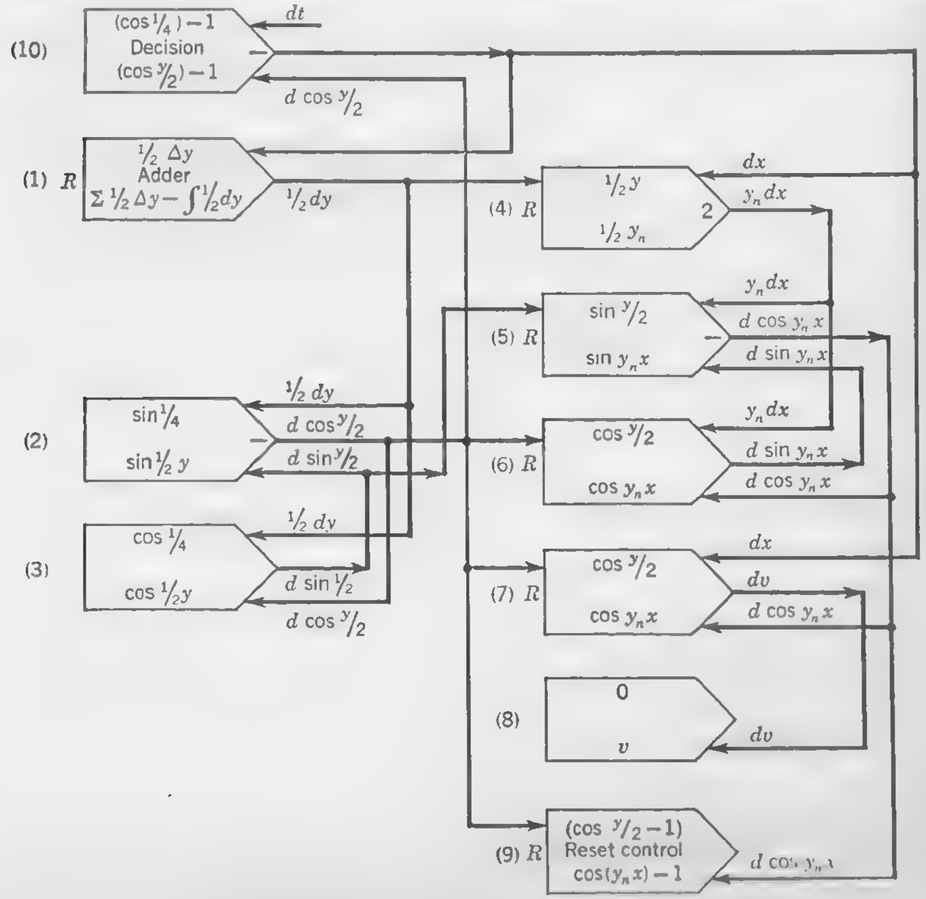


Fig. 19.38. Integrator connections for double integral.

integrator and the effects of these errors on subsequent steps. The single-step errors are the result, on the one hand, of the truncated Taylor series employed in integration and, on the other, of the rounding of Y, R, and ΔZ to a finite number of places.

The integration scheme described on page 19–15, rectangular (Euler) integration, employs two terms in the expansion of $f(x + h)$. The form taken by these terms is a

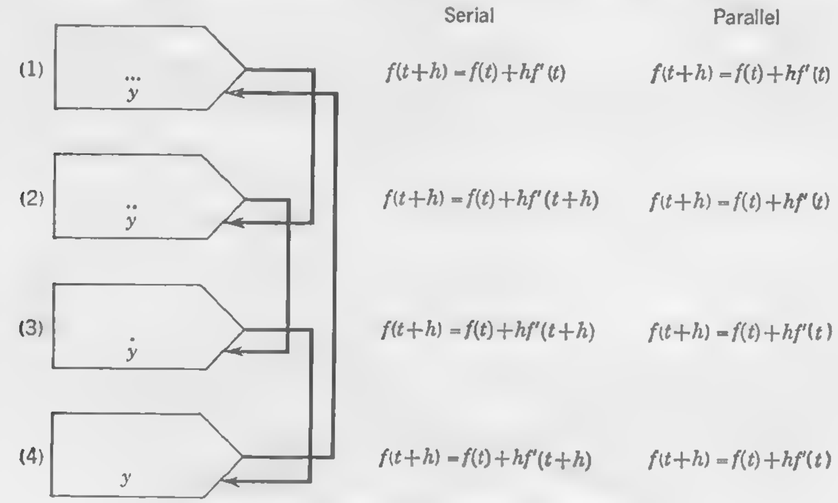


Fig. 19.39. Serial and parallel integration.

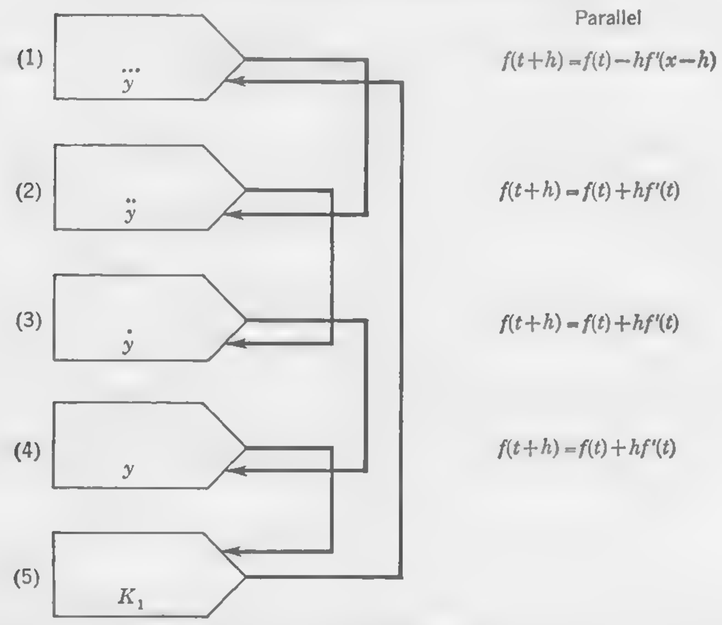


Fig. 19.40. Error with constant multiplier.

function of the position of the integrator in the integration cycle with respect to the integrators that supply its inputs. Figure 19.39 shows the integration formula for each integrator in the serial and parallel solution of

$$\dddot{y} = y \tag{19.10}$$

The magnitude of the error in every integrator in both cases is $h^3/2\ddot{f}(t)$. If, however, Eq. (19.10) is changed to $\dddot{y} = K_1 y$, the delay in the secondary input to integrator 1 in the parallel case, as shown in Fig. 19.40, results in a tripling of the error. In general,

a digital differential analyzer with rectangular integration produces an error of $h^2/2\ddot{f}(t)$ in each integrator provided the secondary input is generated during the same problem increment as the primary input or one increment earlier. Each additional lead or lag introduces an additional error of the order h^2. The derivation of these error terms, it should be noted, entails the assumption that the value of Y employed in the $Y + R$ addition is $Y_{i-1} + \Delta Y_i$. The error generated in integrators that have a primary input that is different from the problem variable, in addition to a variable integrand, is not so easily analyzed but appears to be of approximately the same magnitude. It follows that the numbering of integrators in a serial computer with rectangular integration is arbitrary with respect to truncation error provided that the lags that may be introduced by servos or adders are minimized.

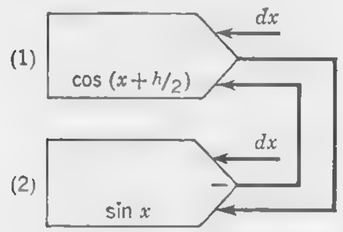


FIG. 19.41. Error in sine-cosine integration.

There is, however, an exception to this rule. In a certain class of problems, errors of the order h^2 may be eliminated from a serial computer by displacing the integrand in selected integrators by half an increment and processing the integrators in a specific order. If, for example, the first integrator in Fig. 19.41 holds cos $(X + h/2)$, the quadrature formula of the sine-generating integrator is

$$f(x + h) = f(x) + hf'\left(x + \frac{h}{2}\right)$$

The cosine-generating integrator, therefore, employs

$$f\left(x + \frac{h}{2}\right) = f\left(x - \frac{h}{2}\right) + hf'(x)$$

and the error term in both cases is $h^3/24f'''(x)$. The same method may be applied to the example in Fig. 19.39 by renumbering the integrators 1, 3, 2, 4, and displacing integrators 1 and 3 by half an increment.

In general, however, errors of order h^2 can be eliminated only by employing higher-order quadrature formulas. Of these, the simplest and the most readily implemented is the trapezoidal rule

$$f(x + h) = f(x) + hf'(x) + \frac{h}{2}[f'(x + h) - f'(x)] \tag{19.11}$$

which has $h^3/12f'''(x)$ as an error term. In both a parallel digital differential analyzer and in those integrators of a serial computer that receive secondary inputs from integrators processed later in the iteration cycle, the $f'(x + h)$ term of Eq. (19.11) is not available. Rather than interpolate, these integrators must extrapolate using

$$f(x + h) = f(x) + hf'(x - h) + \frac{3h}{2}[f'(x) - f'(x - h)]$$

where the error term is five times the interpolative error.

The algorithms for the two modes of operation are:

Interpolative:

$$\begin{aligned} Y_i &= Y_{i-1} + (\Delta Y)_i \\ (Y_d)_i &= Y_{i-1} + \tfrac{1}{2}(\Delta Y)_i = Y_i - \tfrac{1}{2}(\Delta Y)_i \\ R_i &= R_{i-1} + (\Delta X)_i(Y_d)_i - (\Delta Z)_i \end{aligned}$$

Extrapolative:

$$\begin{aligned} Y_i &= Y_{i-1} + (\Delta Y)_i \\ (Y_d)_i &= Y_{i-1} + \tfrac{3}{2}(\Delta Y)_i = Y_i + \tfrac{1}{2}(\Delta Y)_i \\ R_i &= R_{i-1} + (\Delta X)_i(Y_d)_i - (\Delta Z)_i \end{aligned}$$

It should be noted that the subscript i refers to a particular computer iteration cycle whereas the $(x + h)$ of the formulas discussed above refers to a particular step in the independent variable of an integrator. The two may be considered as identical, except perhaps for a scaling factor and an initial offset, only in those integrators which employ the computer independent variable as a primary input.

With $X \neq 0$, the application of these algorithms is straightforward: the old value of Y, Y_{i-1} is read out of the memory, added to ΔY_i, and at the same time, added to either $\frac{1}{2}$ or $\frac{3}{2}\,\Delta Y_i$. The former sum is read back into the memory; the latter is added into R under the control of $(\Delta X)_i$. With $\Delta X = 0$, it appears that an additional register is required to accumulate Y_d until the next nonzero ΔX. This register holds

$$Y_d = Y_{i-1} + \frac{1}{2}\sum_{i}^{i+n} \Delta Y_s \tag{19.12}$$

or

$$Y_d = Y_{i-1} + \frac{3}{2}\sum_{i}^{i+n} \Delta Y_s \tag{19.13}$$

where n is the iteration in which the following nonzero ΔX occurs. At the end of n, Y_d is reset to Y for the next interval. The difference between an extrapolating and

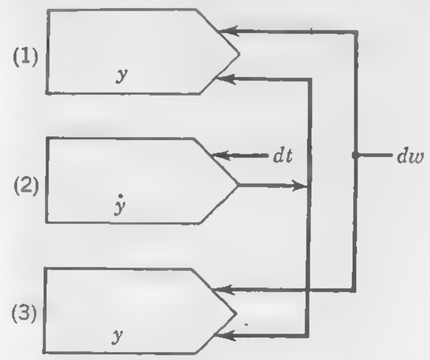


FIG. 19.42. Extrapolation.

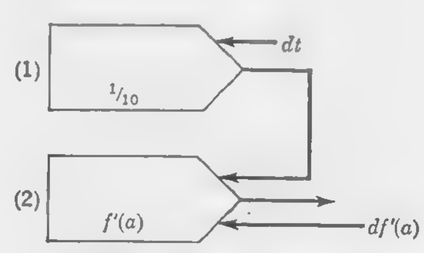


FIG. 19.43. Error reduction.

an interpolating integrator, however, is one computer cycle rather than one step of the problem independent variable. In Fig. 19.42, for example, integrator 1 extrapolates in order to approximate at the start of an iteration the value in integrator 3 at the end of that iteration. The difference between the two, in the example, can be at most one ΔY increment. If, however, Eq. (19.13) is employed in integrator 1, the difference may be greater. Equation (19.13), therefore, should be

$$Y_d = Y_{i-1} + \frac{1}{2}\sum_{i}^{i+n-1} \Delta Y_s + \frac{3}{2}\Delta Y_n \tag{19.14}$$

in order that Y_d in integrators 1 and 3 remain one iteration apart until $dW \neq 0$, at which time the extrapolation occurs.

A further difficulty arises in applying (19.12) to integrator 2 in Fig. 19.43. Because the R register of 1 is rounded off, the first primary input into integrator 2 will occur in a serial computer during the fifth iteration. If integrator 2 employs Eq. (19.10), the number in Y_d at this time is the value of Y at $h/4$ rather than at $h/2$. The resultant error in the output of integrator 2 could be remedied by eliminating the rounding off of integrator 1; the dX input to integrator 2 would then occur when Y_d holds the value of $f'(q)$ at approximately $h/2$. Any output of integrator 2, however, represents an increment of $f(q)$ which should be effective over the entire interval h. If the output

occurs at the end of the interval, a lag is introduced. By modifying Eq. (19.10) to

$$Y_d = Y_{i-1} + \sum_{i}^{i+n-1} dY_s + \frac{1}{2}\Delta Y_n \tag{19.15}$$

both the magnitude and the phase errors are reduced. By the same argument it follows that the extrapolative mode should employ

$$Y_d = Y_{i-1} + \sum_{i}^{i+n-1} dY_s + \frac{3}{2}\Delta Y_n \tag{19.16}$$

In both cases, n is the first iteration after $i - 1$ in which $X \neq 0$. Algorithms (19.15) and (19.16), unlike (19.10), (19.11), and (19.12), do not require a Y_d register. Rather, Y is produced in the normal manner, and whenever $dX \neq 0$, either $\frac{1}{2}$ or $\frac{3}{2}$ is added to the old value of Y and the result is added to R. It should be noted that Eq. (19.15) is identical to the algorithm for multiplication discussed on page 19–32.

In numbering the integrators of a serial digital differential analyzer with trapezoidal integration, the over-all error is generally minimized by minimizing the number of integrators that must extrapolate. In Fig. 19.39, for example, only one integrator, the first, need extrapolate. If, however, the integrators are renumbered 4, 3, 1, 2, integrators 3 and 1 must extrapolate. In general, integrators that receive inputs from lower-numbered integrators interpolate, while integrators that receive inputs from higher-numbered integrators extrapolate. In either case, however, errors of order h^2 are eliminated.

In a parallel computer each integrator extrapolates. Unlike the serial case, the addition of a delay changes the order of the error rather than generating an additional error of the same order. In Fig. 19.40 the error in integrator 1 is $h^2f''(x)$. Errors of this type may be mitigated, at the expense of slowing down the iteration rate, by including a constant multiplier within each integrator. More complex cases require either a further serialization or a complication of the quadrature formulas: The error of integrator 1 in Fig. 19.40 can be reduced to $^{23}/_{12}h^3f'''(x)$ by employing

$$Y_d = Y_{i-1} + \sum_{i}^{i+n-1} \Delta Y_s + \frac{5}{2}dY_n$$

The inclusion of this or more elaborate first-order formulas, or of higher-order algorithms, does not appear to be appropriate in a digital differential analyzer, however, because of the increase in equipment entailed and because of the relative magnitude of truncation and round-off errors (see page 19–48).

Round-off Error. In discussing the arithmetic of binary- and ternary-transfer digital differential analyzers, it has been shown that the round-off error of the former is

$$-1 \leq e \leq +1$$

while that of the latter is

$$-0.5 < e < +0.5$$

In addition to this "normal" round-off error, two related errors arise in a binary-transfer computer because of the lack of a zero-output state. The first of these, phase error, is generated whenever the primary input to an integrator is equal to or near a zero rate. A zero- or near-zero-output rate should result; however, depending upon the relative phase of the ΔY and ΔX inputs, either a positive or a negative error results. For example, if both the primary and secondary inputs are zero rates, one of the two following sequences occurs:

ΔX	1	−1	1	−1	1	−1	. . .
ΔY	1	−1	1	−1	1	−1	. . .
ΔX	1	−1	1	−1	1	−1	. . .
ΔY	−1	1	−1	1	−1	1	. . .

In the former, Y is added to R when it is large and subtracted when it is small so that a positive error results; in the latter, Y is added to R when it is small and subtracted when it is large so that a negative error results. This type of error may be somewhat reduced by using Y_{i-1} rather than Y_i during the ith cycle if $(\Delta X)_i$ is positive and $(\Delta X)_{i-1}$ was negative. However, smoothing of this sort, while effective for a zero-input rate that takes the form of alternating +1's and −1's, introduces errors when applied to other types of zero representations such as

$$+1 \quad +1 \quad -1 \quad -1 \quad +1 \quad +1 \quad -1 \quad -1$$

Without phase errors, the round-off produced in any given iteration is relatively small. But, because a digital differential analyzer generally solves problems by means of a closed loop, this small error in the output of an integrator may affect the integrator input in such a way as to produce large errors quickly. In the solution of $Y' = Ae^x$ where $A = 0$, the output rate of the integrator at the start of the solution is zero while the integrand alternates between zero and either +1 or −1. Assuming that the primary input is computer time, either +1 or −1 will be added to R during every alternate iteration. The resultant erroneous output will accelerate the speed at which the second error occurs, the process continuing until the Y register reaches its maximum value, all of which is erroneous. The effect of this error, called biased round-off, can, like phase errors, be mitigated in some cases by smoothing. Any scheme that is simple enough to be feasible, however, introduces lag. For example, if consecutive plus and minus secondary inputs are canceled, the following input pattern

FIG. 19.44. Round-off error.

	+1	−1	+1	+1	−1
causes	0	0	0	+1	0

to be added to Y, where the +1 is one iteration later. As in the previous case, representations of zero other than alternating +1's and −1's are not filtered.

Of the three types of errors examined thus far, the last two, although independent, are both the result of the lack of a zero-output state and so may be eliminated by a ternary-output system. The first cannot be eliminated since it is a consequence of the finite precision with which integrator outputs are specified. The magnitude of the round-off error can be reduced, however, by increasing the number of output states. Table 19.1 indicates the maximum errors in the seven simplest output schemes together with the number of bits required to store each output and the output values. The latter are assumed to be equidistantly spaced in the interval −1 to +1 since any other choice of values increases the magnitude of the maximum error. As the number of states increases, the round-off error proper decreases. Phase and biased round-off errors are always present if a zero-output state is not included but are generated at a slower rate as the number of states increases.

The implementation of the binary and ternary systems has already been discussed. The latter requires, in addition to the equipment involved in the former, an additional Z line in a serial computer, and twice the number of output lines in a parallel computer. Additional circuitry is also required for output discrimination, output storage, and input registers and counters. Of the remaining schemes, only the quinary is easily implemented, and it is viable only in a computer which employs a binary number system since the product $Y\,\Delta X$ when $\Delta X = \pm\frac{1}{2}$ can be formed in a binary-radix computer by adding Y into R one pulse time earlier than is the case when $\Delta X = \pm 1$. An extra bit of static storage in the Y register recirculation loop is required for reading Y early and an extra place in the R register for holding the half units that may be produced by the multiplication. In addition to this, quinary intercommunication requires a comparable amount of additional equipment as is involved in the step from binary to ternary.

The round-off errors considered thus far are a consequence of the limited precision with which ΔZ increments are specified. Another type of round-off error is generated

by truncating the Y register so that the $Y + \Delta Y$ addition can be initiated with a start pulse (see page 19–19) that is held in the Y register itself. Unlike the former errors, this error is a uniform bias that during any iteration is equal to the value of the discarded digits. If, for example, the initial condition of a variable is rounded off after three digits, the error at each step can be 5×10^{-4} relative to the full-scale value of the Y register. The error in Z, therefore, may be as large as one increment for every 2×10^3 increments of the independent variable.

In many problems errors of this type do not arise because the initial values, or the appropriately transformed initial values, can be represented exactly with only a small number of nonzero digits (zero, "unit displacement," etc.). In other cases, the initial conditions of some of the variables are derived by substitution in the given equation and so have only a limited significance if, as is often the case, the coefficients of the equation are empirically determined. Finally, if the number of digits in the Y register, as determined by the scaling, is large, the error may be small as compared with the other errors.

In those applications in which this error cannot be tolerated, it may be eliminated—within the limits set by the length of the Y register—if the $Y + \Delta Y$ addition is initiated by information stored in another register so that the entire Y register can be employed to hold the initial condition of the variable. Either a pulse in an otherwise empty register or a stored number in conjunction with a counter may be used. A less costly solution that is applicable to a decimal computer is to employ a flip-flop, in conjunction with the flip-flop that normally initiates the $Y + \Delta Y$ addition, in order to move the start pulse two decimal places to the right. The error in ΔZ is thereby reduced by a factor of 100, which is generally within the limits of tolerance.

Because of the prohibitive cost, none of the above solutions is totally applicable to a parallel computer. A compromise is to provide an additional register for the start pulse in a restricted number of integrator units. An alternative, which is applicable to integrators that contain an initial-condition register, is to store the computed value of the integrand and the start pulse in the Y register while the extended initial condition is held in the I register. Additions into R to the right of the least significant digit of the integrand may then employ the number in I while the remainder of the addition employs the number in Y.

The discussion thus far has assumed that the arithmetic employed does not involve scaled multiplication. In the event that it does, the error may be mitigated by moving the point at which the $Y + \Delta Y$ addition is initiated an appropriate number of places to the left. Although this does not in general require any appreciable increase in circuitry, it does necessitate longer registers and less efficient use of storage capacity.

Table 19.1

No. of output states	No. of bits required to store each output	Max error in units of the least significant digit	Values of incremental outputs
2	1	1	1 −1
3	2	½	1 0 −1
4	2	⅓	1 ⅓ −⅓ −1
5	3	¼	1 ½ 0 −½ −1
6	3	⅕	1 ⅗ ⅕ −⅕ −⅗ −1
7	3	⅙	1 ⅔ ⅓ 0 −⅓ −⅔ −1

Total Error. In designing a digital differential analyzer, an estimate of the relative importance of the factors that contribute to the total error is required in order to arrive at an optimum combination of integrating and rounding-off processes. Unfortunately a complete theory of the generation of error in a digital differential analyzer does not exist.* However, certain heuristic considerations may be employed which, on the whole, are substantiated by computational experience.

Each variable in a digital differential analyzer is held jointly by two registers (see Fig. 19.45), a Y register and the corresponding R register that supplies its input. (Essentially the same analysis may be applied if several R registers provide inputs.) The single-step round-off error in any variable is generated in the least significant digit of the R register. The rounding off of Y has no effect upon subsequent computations except in so far as it generates errors in the R register to which Y is added. But this error, because Y is multiplied by h before being added to R, has the value of the least significant digit of the succeeding R register. Assuming that the round-off errors have a random distribution, the expected error at each step with binary transfer is $0.58S$, where S is the machine value of a unit in the least significant digit of R. Because S equals h^2 on the average, after N steps the cumulative error is $0.58h^2N$. This estimate, however, applies only to those integrators which employ the computer independent variable as a primary input; i.e., it applies essentially to the solution of linear equations. If another integrator provides the primary input, the size of the round-off error is substantially increased. The error expression for a single iteration in this case, where $\epsilon = \pm 0.58h$, is

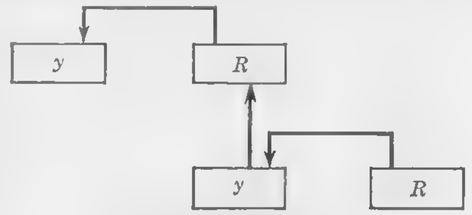


FIG. 19.45. Total error.

$$(Y + \Delta Y + \epsilon)(\Delta X + \epsilon) = (Y + \Delta Y)\Delta X + \epsilon\,\Delta X + (Y + \Delta Y)\epsilon + \epsilon^2$$

rather than, as in the linear case,

$$(Y + \Delta Y + \epsilon)\Delta X = (Y + \Delta Y)\Delta X + \epsilon\,\Delta X$$

Ignoring second-order terms, the error is $\pm 0.58hY$. To estimate Y in the general case, however, is difficult since it depends upon both the function Y and the efficiency with which the problem is scaled. A conservative estimate, considering that Y is generally biased in one direction, is probably $\frac{1}{10}$ so that the round-off error after N steps is $\pm 0.058hN$ or $h^{-1}/10$ times the error in the linear case. If either a servo or an adder supplies the primary input, this error increases with the number of secondary inputs to the servo or adder. The number of these inputs may in effect be decreased by performing all or some of the required additions in a multi-integrand integrator similar to that described on page 19–32.

The truncation error for rectangular integration is $h^2/2$ at each step assuming that $f''(t)$ can be ignored. After N steps, the error is of the order of $Nh^2/2$ for a large class of problems since, unlike round-off errors, truncation errors generally tend to be biased in one direction. In certain cases a canceling effect may reduce the error, while in others instabilities may cause the error to build up more rapidly. As N increases, therefore, the truncation error in a binary-transfer digital differential analyzer with rectangular integration very often becomes large relative to the round-off error in the linear case. If the number of nonlinearities is large, the two errors probably tend to be more nearly equal.

With the introduction of trapezoidal integration, the magnitude of the truncation error becomes approximately $Nh^3/4$. Assuming ternary transfer, in the linear case

* Estimates of the propagated error due to truncation in the generation of simple functions may be derived by finding the roots of the related difference equations. For a discussion of an analogous but more generally applicable technique that relates the stability of the solution of a linear differential equation to the integration formula and the interval of integration, see Ref. 26. In Gray's terminology, rectangular integration represents a 0_{11} formula in a parallel computer or, in a serial computer, either a 0_{11} or a form of an $[0_{11}C_{11}]$ formula depending upon the position of the integrator with respect to its inputs; extrapolative trapezoidal integration employs 0_{11} and interpolative trapezoidal integration employs an algorithm similar to $[0_{11}, C_{11}]$.

the truncation error is larger than the round-off error if $h^2 > N^{-1}$ and is smaller if $h^2 < N^{-1}$. With $h = 10^{-3}$, for example, the two errors are approximately equal after 10^6 iterations or about 3 hr of computation on presently operative serial equipment. This time increases as the inverse square of h so that an h of 10^{-5} would require 3×10^4 hr for the truncation errors to reach the magnitude of the round-off errors. For a high-speed parallel computer this would require 10 or 15 hr. Most problems, however, fall well within the range of h and N in which the truncation error is considerably smaller than the round-off error, particularly if nonlinearities are involved. In these cases, a reduction of the round-off error by a factor of r increases the speed of computation by a factor of r. That this is the case may be seen by considering a digital differential analyzer that employs a binary radix and binary intercommunication. The last digit in every Y and R register can be predicted from the number of secondary inputs to the integrator and the parity of the iteration without reference to the functions being generated. In fact, if the number of secondary inputs to all integrators is even, every register can be reduced by one bit without affecting the computation. This bit is regained in a computer with ternary transfer and an additional bit is generated with a quinary system. The accuracy of ternary intercommunication is thus twice that of binary while quinary transfer is twice as accurate as ternary. If the effective precision in the three cases is kept equal, the gain in accuracy may be translated into a gain in speed. This method of increasing the speed of computation—where it is applicable—is in general considerably less expensive than other methods, such as increasing the pulse-repetition rate.

From the foregoing discussion it should be clear that the choice of integration and intercommunication schemes is interrelated. An optimal design within the limitations imposed by the application is one which tends to equalize the two types of error. Assuming trapezoidal integration, for example, this is very often the case if quinary transfer is employed in a serial computer or if ternary transfer is employed in a parallel computer. Any further significant reduction in the total error requires a reduction in both truncation and round-off errors. The reduction of either, however, necessitates the introduction of a digital multiplier and so entails sacrificing the basic simplicity of the digital differential analyzer conception. In one sense this does not constitute a basic limitation of the digital differential analyzers, however, since the total error can always be reduced by decreasing the interval of integration.

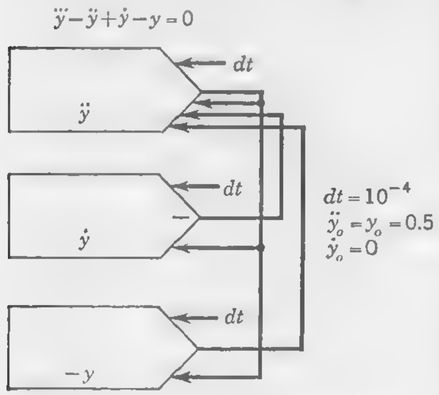


FIG. 19.46. Example for truncation error.

The magnitude of the total error in the solution of a given problem on a specific computer may often be estimated by solving the problem, reversing the sign of the independent variable, running the solution in the reverse direction until the initial condition of the independent variable is reached, and comparing the resultant values of the dependent variables with the initial conditions. This procedure is facilitated if a means for reversing the sign of the computer independent variable—normally under control of a "time" switch on the control console—is provided. In a serial computer this may be provided by modifying the term that inserts time onto the Z line (see page 19–22). In a parallel computer that employs a plugboard, the switch can operate directly to change the time source. The primary input to servos and adders, which is normally plus time, must, however, be independent of the position of the time switch. More orthodox methods for estimating the total error, such as varying the interval of integration, are also applicable.

Numerical Examples. The discussion of error in the three previous sections may to some degree be verified by numerical examples. The solution of $\dddot{y} - \ddot{y} + \dot{y} - y = 0$ using the schematic in Fig. 19.46, for example, illustrates the effects of truncation

Table 19.2

t	Error × 10^4			
	Serial		Parallel	
	Rectangular	Trapezoidal	Rectangular	Trapezoidal
$\pi/2$	−1	−1	0	0
π	−3	0	−1	0
$3\pi/2$	−15	0	−1	−1
2π	−71	−3	−2	−3
$5\pi/2$	−338	−16	−17	−15
3π	−1,625	−73	−82	−73

errors on an unstable solution. The errors in y resulting from rectangular and trapezoidal integration (both with ternary transfer) are shown in Table 19.2. The surprisingly small error in the parallel case with rectangular integration results from a phasing relationship similar to that discussed on page 19–42. Note that by employing a configuration different from that in Fig. 19.46 the same results could be obtained in

Table 19.3

t ···	Error in cos t			
	$E \times 10^2$		$E \times 10^3$	
	Rectangular	Trapezoidal	Rectangular	Trapezoidal
10π	9	0	4	0
20π	18	−1	8	0
30π	30	−1	11	0
40π	44	−1	16	0

Table 19.4

x	$\frac{\text{Error}}{e^x} \times 10^4$			
	Binary rectangular	Transfer trapezoidal	Ternary rectangular	Transfer trapezoidal
0.5	−0.29	0.23	−0.84	−0.35
1.0	−0.89	0.21	−1.36	−0.23
1.5	−1.46	0.22	−1.99	−0.24
2.0	−2.16	0.35	−2.63	−0.26
2.5	−2.78	0.34	−3.22	−0.24
3.0	−3.42	0.34	−3.82	−0.25
3.5	−4.06	0.37	−4.42	−0.23
4.0	−4.66	0.38	−5.04	−0.24

the serial rectangular case. In general, however, a reduction in error of this type is not possible. A further comparison of the effects of integration formulas upon error is shown in Table 19.3, which gives the error in the parallel generation of cos t at two scale factors. Note that a reduction in the step size by a factor of 10 results in a reduction in the error by a factor larger than 10—as should be expected from the previous analysis. A third example, the generation of e^x, is given in Table 19.4.

Representative examples for the comparison of various intercommunication schemes are difficult to present because of the statistical nature of the error. In

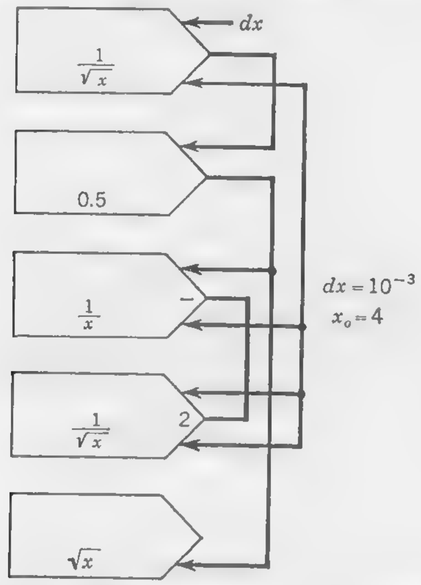


FIG. 19.47. Error dependent on transfer scheme.

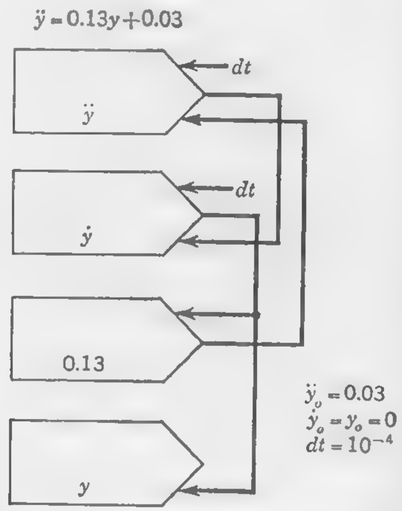


FIG. 19.48. Round-off due to extended zero rates.

Table 19.4, for example, a comparison of the trapezoidal cases indicates the advantage of ternary transfer; however, in the rectangular case binary transfer produces a smaller error. This may be either a statistical phenomenon or, perhaps, a result of the interaction of round-off and truncation error. In some cases, however, the advantage of ternary transfer should be predictable. For example, in integrator configurations in which, as in Fig. 19.47, many integrators employ a variable other than time as a primary input, the error should be strongly influenced by the transfer scheme. This can be seen in Table 19.5.

Table 19.5

x	Error × 10^3					
	Binary			Ternary		
	$\frac{1}{\sqrt{x}}$	$\frac{1}{x}$	$\sqrt{x}$	$\frac{1}{\sqrt{x}}$	$\frac{1}{x}$	$\sqrt{x}$
24	1	0	8	0	0	−2
44	0	1	20	−1	0	−6
64	0	−2	33	−1	0	−13
84	1	0	48	−1	0	−21

Relatively large round-off errors should also be produced by a binary-transfer computer in solutions which involve zero rates for extended periods. This is the case in the solutions of $\ddot{y} = 0.13y + 0.03$ by rectangular integration shown in Fig. 19.48 and Table 19.6.

Table 19.6

t	Error × 10^4	
	Ternary	Binary
1	0	1
2	0	4
3	0	10
4	0	18
5	0	31
6	−1	49

Thus far the binary solutions that have been considered have employed the ΔX smoothing scheme described on page 19–46. The magnitude of the error that results without smoothing is suggested by a comparison of Table 19.7 and the first error column of Table 19.4. The former shows the results of generating e^x without smoothing when the pattern of the independent variable is a series of 16 +1's and 48 alternating +1's and −1's. The latter illustrates the results using continuous +1's. If ternary transfer is used, the results in the two cases are identical.

Table 19.7

x	Error / e^x × 10^4
	Binary rectangular
0.5	−5.70
1.0	−10.18
1.5	−13.21
2.0	−13.74
2.5	−15.89
3.0	−18.79
3.5	−21.73
4.0	−24.10

The two final examples, shown in Figs. 19.49 and 19.50, are included to illustrate the accuracy of a digital differential analyzer in solving nonlinear problems of some complexity. The first, Mathieu's equation, was run for one cycle with no error. The second, an equation common to fluid-flow problems, shows an occasional error of 1 in the least significant digit for values of t up to 4.

19.3.5. Input and Output. *Initial-condition Input.* Two types of information must be entered into a digital differential analyzer—initial conditions, including addresses in a serial computer, and functions. Initial conditions may be entered manually from a keyboard, automatically from a tape or punched cards, or, in a special-purpose computer, by some form of wiring such as a plugboard. Provisions for manually altering information are generally included in computers that are normally filled automatically.

The first step in the manual filling operation is to specify—by a switch setting, usually—the integrator to be operated upon. In a serial computer this switch normally operates in conjunction with the integrator counter to indicate the desired integrator time. In a parallel computer the switch connects the specified integrator to the remaining input circuitry. Once an integrator has been selected, the integrand may be entered from a numerical keyboard by being precessed into the correct position, one or several digits at a time. In this case negative numbers are generally entered in complementary form. An alternate scheme that involves less control circuitry employs a pulse, called a "marker," which is entered into one of the registers that does not receive initial conditions—the Z line, for example, or the Y register if initial conditions are held separately. As each digit is entered, it is positioned by the

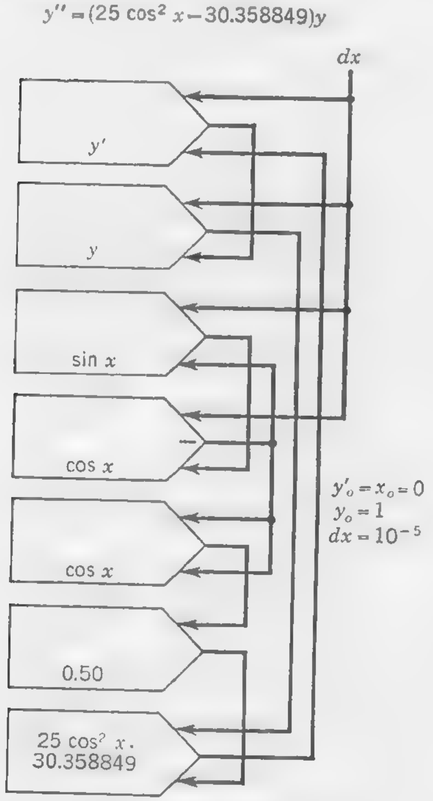


Fig. 19.49. Mathieu's equation.

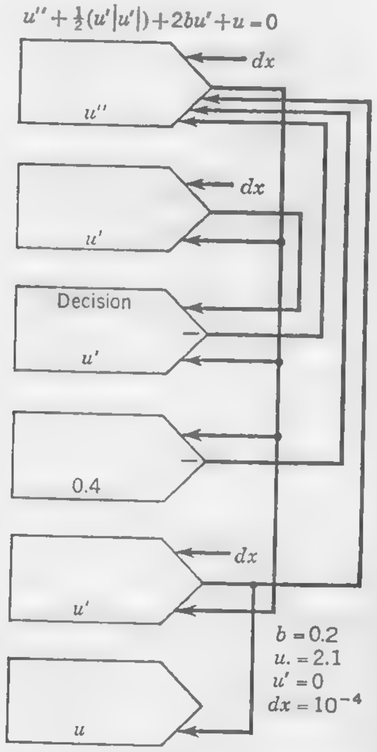


Fig. 19.50. A fluid-flow problem.

marker in the register that is being filled and the marker is moved, either manually or automatically, the requisite number of pulses to the right. The same circuitry may be employed in a serial computer for entering addresses.

Automatic filling schemes can be approached in two ways—with a view to minimizing the amount of equipment or to maximizing the convenience of operation. The former requires that the information entered on the tape have the same configuration as the information which is to be placed in the storage medium. In a serial computer this necessitates the explicit development of the entire drum configuration before the tape can be prepared. Figure 19.51 shows a drum section and the corresponding tape. As in the manual case, either a marker or precession may be used to position the information correctly in the computer.

If operational convenience is to be maximized, integrands should be entered as sign and absolute value. Once in the computer, however, the necessary complementation is an awkward process. An alternative is to generate and enter the $P - 1$ complement (where P is the radix) of each digit as it is read if the sign of the number is

negative. The resultant error can be corrected by increasing the absolute value of negative numbers by one before placing them on the tape. With this scheme, the additional equipment required, beyond that needed for processing tapes which contain complements, is little more than the single bit of static storage which, by holding the sign of the integrand, controls complementation.

In addition to integrand values, a serial computer also requires addresses. To require that the positions of addresses be computed and their pulse positions entered is inconvenient. In a computer with a binary integrator output scheme there is no simple device by which this inconvenience may be removed, unless integrator outputs are stored statically. In a ternary-output computer one Z-line may be used during the filling process to hold artificially produced outputs of all integrators with the same first digit as the address, while the other Z line holds the outputs of those integrators with the same second digit. Each integrator will thus contain one and only one pulse time in which a pulse occurs in both of the two Z lines—at the position where the address integrator occurs. At this point in the integrator being filled, a 1 is entered in the appropriate address line and the Z lines are cleared for the next address. If integrator 38 is to be a secondary input for integrator 72, for example, as the 3 is

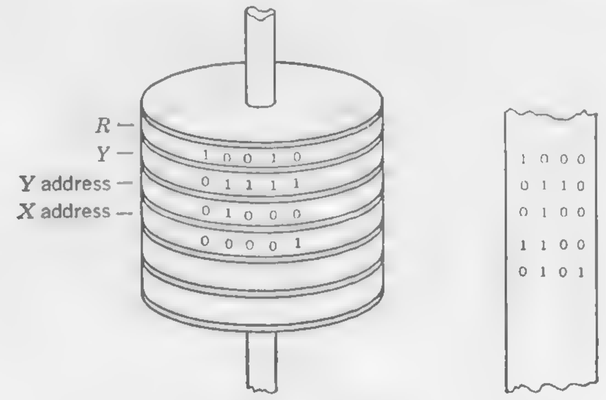


Fig. 19.51. Input tape.

read from the tape, integrators 30, 31 . . . 39 produce outputs which are recorded on the Z-existence line. As the 8 is read, integrators 08, 18, 28 . . . 98 produce outputs which are recorded on the Z-sign line. Only the position corresponding to integrator 38 will contain a 1 in both Z lines and at this point a 1 is entered in the Y-address line associated with the Z reading head in which the coincidence occurred. For digital differential analyzers with more than one hundred integrators a third line is required in order to implement this system.

Function Input. The solution of certain problems by a digital differential analyzer requires the employment of functions that are not generated by the computer itself. These may be empirical functions which do not have a simple analytical representation or analytic functions which are difficult to generate. Three of the devices that may be employed to enter functional values into the computer are an analog-to-digital converter, a digital graph follower, or a tape reader.

An analog-to-digital converter may be used as an input device when a digital differential analyzer is a portion of a control system, or a portion of a physical simulator. A converter is also required if an analog graph follower supplies the input function. In all these cases the input normally takes the form of incremental changes in the function rather than that of functional values proper. These increments, in a serial computer, can be placed directly on the Z line and used as if they were integrator outputs. In a parallel computer the information is patched into the plugboard, if one is used. In both cases the signals can be synchronized either by placing the inputs into static storage as they occur and synchronizing the time at which the storage devices are read or by providing the converter with signals from the computer which synchronize the converter outputs. The inputs from a digital graph follower

or an incremental tape are handled in essentially the same manner. The latter device holds the function as a series of signed unit increments with an additional channel serving to indicate the portion of the tape which refers to a given interval of the independent variable.

In using an incremental function input, a digital differential analyzer that does not have a zero transfer state must generate a zero rate whenever information is not being received from the input device. A single flip-flop may be used for this purpose which changes state each iteration but is inhibited from changing state in one direction by a positive input and in the other direction by a negative input. Since a given input may be required to wait as long as two iterations for the proper pattern of change to occur, inputs may not be accepted at the computer iteration rate.

It often is more convenient to process a function by using functional rather than incremental values. This permits the generation of oddly formed step functions, but, more importantly, it permits differences that have a magnitude greater than

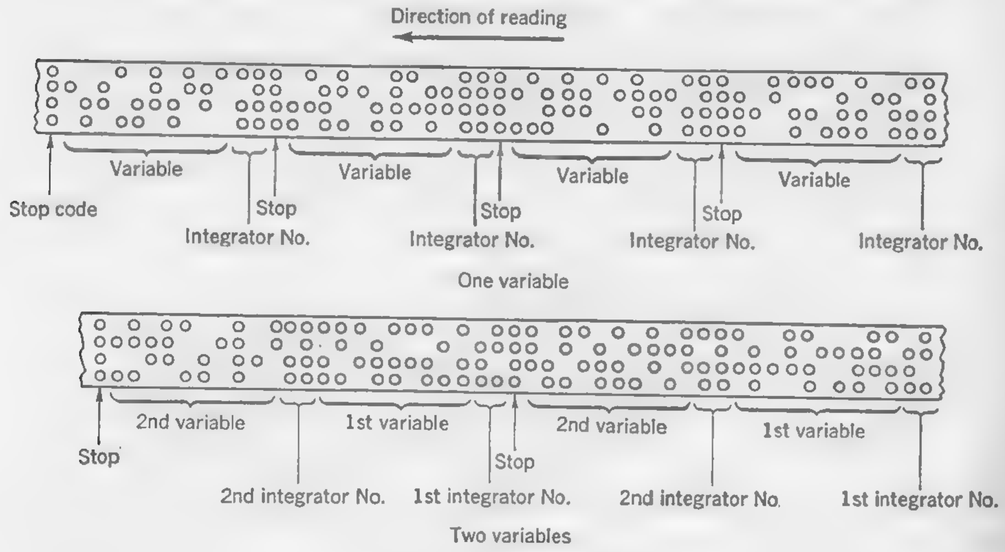


FIG. 19.52. Punched tapes.

1 to be employed. These differences may be entered and operated upon internally in order to supply the required function. Not only is the preparation of tapes made less arduous but the speed at which functions may be entered is substantially increased. Using this system, a slow paper-tape reader is often faster than a fast graph follower. Moreover, the equipment which is used to process initial conditions can, with few changes, also serve to process one or more input functions, since a single reader may process several input functions provided all are controlled by a single variable. Figure 19.52 shows tapes for one and two functions. The value of the independent variable, it should be noted, does not appear on the tape. This variable is employed within the computer to control the point at which the value of the input function is changed. Magnetic tapes can be employed if higher input rates are required.

Schemes for employing zero-, first-, and second-order differences are illustrated in Fig. 19.53. The last of these assumes that $X_{n+1} - X_n = 1$; therefore,

$$\Delta^2 Y_n(X_{n+1} - X_n) = \Delta Y_{n+1} - \Delta Y_n$$

so that integrator 3 holds

$$\Delta Y_n + \Delta Y_{n+1} - \Delta Y_n = \Delta Y_{n+1}$$

at the end of the nth interval. This eliminates the necessity for entering the first

difference. Summing the outputs of integrators 3 and 4 gives the Gregory-Newton interpolation approximation.

The amount of time that computation must be halted for the reading of functional values from tape may be substantially reduced by reading into an initial-condition register rather than directly into the Y register. Using this scheme, computation need be halted only for transferring between registers since the next value can be

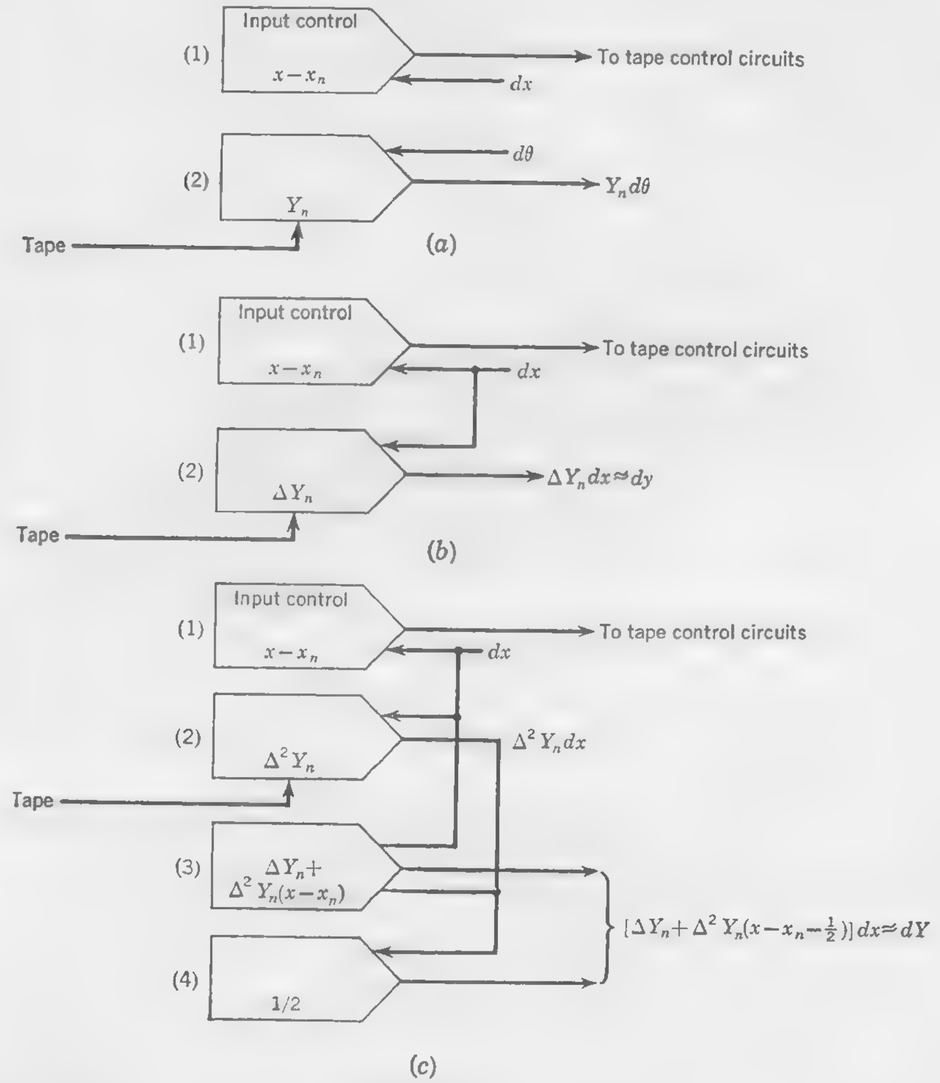


FIG. 19.53. Tape control.

read from the tape after computation has been restarted. More than a single variable may be stored on a tape provided all the variables are functions of the same independent variable. The reading of the next function value may be controlled in many ways: a specified integrator producing a single, or a double output; or the integrand itself reaching a certain value, or exceeding 1. This last scheme is employed in Fig. 19.53. If functions of more than one independent variable are employed, multiple-tape units are required.

Thus far it has been assumed that the independent variable controlling the movement of the tape through the reader is monotonic. If it is not, the tape must reverse direction whenever the derivative of the independent variable has reversed in sign

and the variable itself has reached the edge of an interval. With Y_n in the Y register and Y_{n+1} in the initial-condition register, upon reversal the tape must be read past Y_{n+1} and Y_n and then must read Y_{n-1} into the initial-condition channel. This may be accomplished either by reading past Y_{n-1}, reversing, and reading the value in the forward direction, or by reading directly in the reverse direction. The former scheme requires that the tape be moved three times for each functional value when operating in the reverse direction while the latter necessitates the modification of the read-in equipment to facilitate correctly positioning values which are read from the least significant end. In addition, since the number of the integrator that is to receive the information and the sign are read after the function value when the tape is operating in the reverse direction in the latter case, negative numbers must be placed on the tape as complements and the number of the integrator into which the values are to be placed must be specified by the wiring of the computer.

Printed Output. The printing of results in a digital differential analyzer is generally controlled by a single variable which produces an overflow in either a Y or an R register. The latter is a more flexible system since intervals are not restricted to integral powers of the radix. For example, if answers are to be printed every $\pi/2$ increments of the variable dQ, the scheme in Fig. 19.54 may be used. By manually or automatically changing the condition of Y register at the start of an interval, results may be produced at uneven intervals of the controlling variable. Furthermore, several integrators may be used to control printing.

In a serial computer, a bit in a specified position of one of the address lines normally indicates which integrands are to be printed. In order to eliminate the necessity for interrupting computation during the printing process, an additional channel—called a T channel—may be employed. The selected integrands are read into this channel, and printing proceeds along with computation. If the interval between print signals is too short to allow the T channel to be processed before another set of values is ready, computation must be halted until the new set of values can be read into T, and then automatically restarted. In a parallel computer, a similar scheme may be employed, with a patchboard serving to select the proper integrands. Depending upon the requirements of the system, information may be sent to the printing device either serially or in parallel.

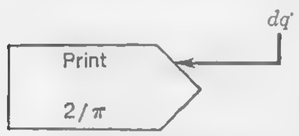


FIG. 19.54. Print control.

Once a print signal has been received, negative numbers are normally recomplemented before being printed. This requires that the sign be held statically for one iteration so as to allow complementation to proceed from the least significant end of the number. Complementation may be accomplished by taking the P-1 complement of each digit after the first nonzero digit and the P complement of all prior digits. In a binary system, for example, all digits up to and including the first one are unchanged and the remaining digits are changed from 1 to 0 and from 0 to 1.

In addition to printing numbers as absolute value and sign, it is convenient to print the results of computation in decimal. If computations are performed using another radix, a conversion must be performed not only on the answer but on the scale factor as well. This latter conversion may be performed by a constant multiplication within the computer. For example, if, in a binary computer, a variable Y is to be printed and the scale of dY is 2^{-6}, multiplying dY by $2^6/10^2$ will result in a scale factor of 10^{-2}. The conversion of the resultant number to decimal, however, cannot be performed in this manner. For this conversion additional circuitry is required. One straightforward approach is to collect the dY increments in a register that has a decimal adder on its input. When type-out is signaled, four bits of static storage are successively inserted into the register recirculation path and the contents of these four bits printed. On the other hand, if the register is completely static, a complete number may be printed at one time. In either case, a register and an adder are required for each variable that is to be printed.

An alternate approach that requires only a single arithmetic unit to process any number of integrands is to convert each integrand in turn. The conversion is accomplished simply by successive multiplications of the integrands by ten (1010). After each multiplication the resultant decimal digit can be sent to the printer. If the number of integrand values to be printed at each data point becomes large, it may be necessary in a parallel computer to provide parallel conversion circuits. This is not excessively costly since each multiplier is essentially a binary adder with provisions for shifting the multiplicand.

Other Outputs. In addition to producing printed copy a digital differential analyzer can be used to plot curves and to produce tapes. These tapes may, of course, merely duplicate information that is normally printed. More importantly, however, they may hold functions to be used subsequently as inputs to the computer. This type of output, which generally does not require radix conversion, can take the form of increments or whole numbers. In the former case, Z-line information is sent to a punch in much the same way as it is sent to control incremental input, except that increments of two variables must be punched. In the latter case, the printing circuitry serves to supply information to the punch.

The results of computation may also be plotted digitally or, using a digital-to-analog converter, analogwise. The latter scheme is particularly appropriate to a parallel computer which produces answers several thousand times as fast as the fastest digital plotter. An even faster output medium for a parallel computer is an oscilloscope. Here, too, a digital-to-analog converter is required, but in addition, if the trace is to be stationary, the computer must have the ability to recycle its contents automatically. Because of its low accuracy an oscilloscope is more appropriate for monitoring solutions than for holding results which are to be photographed. The shape of the trace, however, gives a quick and convenient indication of the form of the solution and the effect of parameter changes. Once the correct parameter values have been approximated, the solution may be accurately read on one of the other output devices. Finally, a single multiplexed digital-to-analog converter or number of individual converters can serve to couple the computer to other devices for control, simulation, or other purposes.

19.3.6. Over-all Computer Design. *Number Systems.* Thus far the component parts of a digital differential analyzer have been described. The specific form that these assume in any computer depends to a large extent upon certain over-all design considerations. One of these is the number system. A computer that employs a binary radix generally requires that input information be in binary form or, at the least, that scaling factors be powers of 2. The inconvenience entailed in handling binary numbers, however, must be weighed against the computation efficiency (see page 19–29) and the equipmental simplicity of a binary computer. Certain applications, in which program changes are infrequent and the output either is very limited or is to be transformed into analog form, gain little from employing a decimal radix. Such applications include process-control computers, large-scale simulation devices, and many special-purpose applications. On the other hand, if a digital differential analyzer is to be employed for general computations, the added cost of a decimal computer—particularly if it is serial—is compensated for by savings in the time required for both the preparation of input data and the interpretation of results. Moreover, the cost of providing a binary serial digital differential analyzer with decimal input and output outweighs the cost of providing decimal arithmetic since, unlike a general-purpose computer, the arithmetic section proper of a digital differential analyzer cannot perform conversions. Because of the multiplicity of arithmetic units, a parallel decimal computer appears to be prohibitively costly. Decimal output in the parallel case, however, is not overly costly provided the internal number system employs fractions—as has been assumed above—rather than integers. Except for this consideration, the internal arithmetic may be constructed so as to operate with the radix point at any position. If the radix is other than binary, the encoding scheme should be self-complementing and such that addition requires only binary adders. Both the excess-deficit-three and the 2-4-2-1 codes have these characteristics.

Register Configuration. The number system having been determined, the number and type of registers are the next consideration. At a minimum, a serial computer requires two channels: a single channel that holds both arithmetic and address infor-

mation, and a Z line. This configuration (see Fig. 19.55), furthermore, permits a reduction in the circuitry of the address unit since the ΔY and ΔX counters can serve as registers during the computational portion of the integrator. If R and Y information are held separately, an additional channel is required; if address and arithmetic information are separated, one or several additional channels are required for addresses depending upon the number of integrators and the universality of permissible integrator interconnections. Additional address channels are also required to implement the more complex arithmetics which follow.

If the number of multiplications to be performed in a given application is relatively large, the addition of a second Y register, as described on page 19–32, results in a substantial increase in computer capacity with a relatively small increase in equipment.

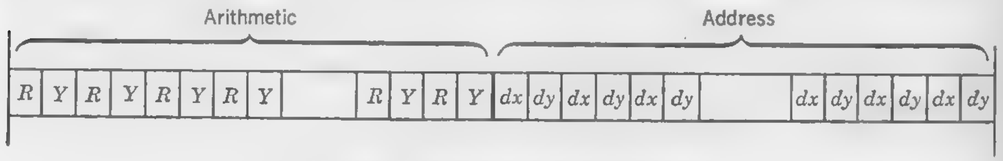


FIG. 19.55. Minimum serial computer.

An extension of this scheme to three or even four Y registers may, in special cases, represent an optimal scheme. On the other hand, additional registers may provide parallel integrators in order to increase the speed of computation or the capacity of the computer. Several of the schemes for multiplication by a constant discussed on page 19–29 also require additional channels. The simplest of these, the scaling multiplier, employs a single additional register and is applicable only in a computer with binary intercommunication. A more general solution employs two additional registers. Additional channels are also required for storing initial conditions and for processing information to be printed if computation is not to be interrupted.

A consideration equally as important as the number of arithmetic channels is the number of adders and associated equipment entailed by a given scheme. In this respect it is often advantageous to provide a single serial-parallel adder rather than

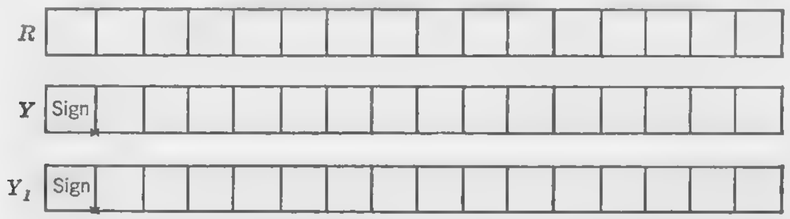


FIG. 19.56. Binary-trapezoidal DDA with initial condition channel.

a number of serial adders and to interlace the serial arithmetic registers among the arithmetic channels. For example, in a computer employing a binary radix, trapezoidal integration, and an initial-condition channel, the normal arrangement of arithmetic information as shown in Fig. 19.56 requires three adders which generate

$$Y + \Delta Y$$
$$Y + \tfrac{1}{2} \text{ or } \tfrac{3}{2}\Delta Y = Y_d$$
$$Y_d + R$$

By rearranging the information as in Fig. 19.57, a single serial-parallel adder is sufficient for performing all the required arithmetic since the Y_d addition may be performed during the reading of Y_I. The wisdom of this design, however, is questionable since an octal serial-parallel adder probably requires more circuitry than three binary serial adders. What Fig. 19.57 illustrates is a method of trading speed for memory capacity in a binary radix computer. For example, by adding a memory channel and employing a sexidecimal serial-parallel adder, the speed of computation and the number of integrators can be increased by one-third. On the other hand, with two

channels and a quaternary serial-parallel adder, the speed and the number of integrators can be decreased by approximately one-third. Either output multipliers or facilities for initiating the $Y + \Delta Y$ addition at any position within a parallel digit must be provided if the scaling efficiency of the completely serial design is to be maintained.

The parallel-serial scheme just outlined is generally more appropriate to a computer with a decimal radix since a single serial-parallel decimal adder requires less circuitry than four serial decimal adders. The following examples indicate the application of the scheme to two of the many possible integrator configurations. A decimal

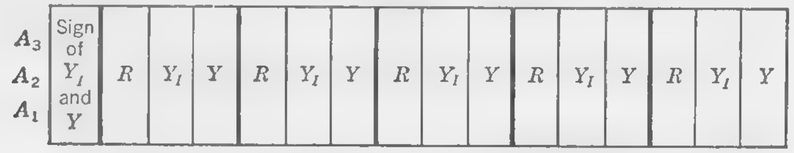


FIG. 19.57. Arrangement with a single serial-parallel adder.

computer that employs trapezoidal integration and a constant multiplier for each integrator requires four registers: R, Y, R_k, and k, and four additions:

$$Y + \Delta Y$$
$$Y + \tfrac{1}{2} \text{ or } \tfrac{3}{2}\Delta Y = Y_d$$
$$R + Y_d$$
$$R_k + k$$

A serial-parallel adder would in this case not only be more economical than the completely serial scheme but could, with a 20 per cent decrease in speed, provide for processing a variable initial-condition register. Furthermore, only a single time-shared output discriminator need be provided whereas the wholly serial configuration of information requires a separate discriminator for each R register. In the event that two Y registers and a scaling multiplier are employed, each integrator must perform five additions:

$$Y_1 + \Delta Y_1$$
$$Y_2 + \Delta Y_2$$
$$Y_1 + Y_2$$
$$(Y_1 + Y_2) + S$$
$$(Y_1 + Y_2 + S) + R$$

Since only four quantities, Y_1, Y_2, S, and R, need be stored, the fifth addition time may be employed to store Y_I.

It has been assumed thus far that the integrator length in a serial computer is fixed, being determined by the application. One exception to this assumption is the split-integrator scheme introducedon page 19–29. The assumption may be dropped by introducing an additional channel in which the beginning of each integrator is indicated by a pulse. This pulse may also serve to initiate the $Y + \Delta Y$ addition if the resultant round-off error can be tolerated. The system, however, is generally not feasible because it requires both static storage for integrator outputs and a complex addressing scheme. A compromise solution for applications that require some flexibility in integrator length is to permit an integrator to operate over multiples of some basic integrator length. This is compatible with the conventional serial output distribution and address systems and may be implemented by placing a bit at the most significant end of the address lines of an integrator to indicate that the addition of Y to R is not to be interrupted.

19.3.7. Programming. *Mapping.* The mapping of a problem for a digital differential analyzer consists of developing a set of integrator interconnections which correspond to the equations to be solved. Several such maps have already been presented. A more generalized statement will now be given.

When the problem is the solution of a single nth-order differential equation,

$$f\left(U, V, \frac{dV}{dU}, \cdots, \frac{d^nV}{dU^n}\right) = 0$$

the first step, generally, is to solve for the highest derivative,

$$\frac{d^nV}{dU^n} = g\left(U, V, \frac{dV}{dU}, \cdots, \frac{d^{n-1}V}{dU^{n-1}}\right) \tag{19.17}$$

At this point one of two methods may be employed to arrive at an equation in differential form. The first takes the derivative of both sides of Eq. (19.17) and yields

$$d\frac{d^nV}{dU^n} = dg\left(U, V, \frac{dV}{dU}, \cdots, \frac{d^{n-1}V}{dU^{n-1}}\right) \tag{19.18}$$

The second transforms (19.17) into

$$\frac{d^nV}{dU^n}\,dU = d\frac{d^{n-1}V}{dU^{n-1}} = g\left(U, V, \frac{dV}{dU}, \cdots, \frac{d^{n-1}V}{dU^{n-1}}\right)dU \tag{19.19}$$

by what may be considered as multiplying both sides of the equation by dU.

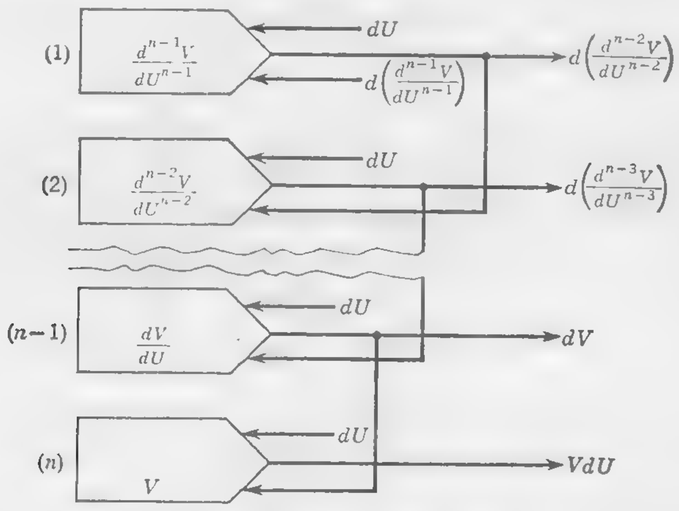


FIG. 19.58. Successive derivatives.

The mapping of Eq. (19.17) is shown in Fig. 19.58. The first integrator holds $d^{n-1}V/dU^{n-1}$ and is integrated with respect to U to form

$$\frac{d^{n-1}V}{dU^{n-1}}\,dU = d\frac{d^{n-2}V}{dU^{n-2}}$$

With this as its secondary input, integrator 2 holds $d^{n-2}V/dU^{n-2}$. Again, integration with respect to U yields the differential of the next lowest derivative and the process is repeated until integrator n holds V. The outputs of these integrators are then combined to form

$$g\left(\frac{U, V, dV}{dU}, \cdots, \frac{d^{n-1}V}{dU^{n-1}}\right)dU$$

which becomes the secondary input of integrator 1. The mapping of Eq. (19.18) proceeds in an identical manner, with the exception that an additional integrator is employed to hold d^nV/dU^n. The choice of methods depends upon the function g;

however, the method of Eq. (19.19) is generally, but not always, more efficient both because the order of the equation is reduced by 1 and because the generation of products is simplified. Developing $UVW\,dt$, for example, requires only three integrators whereas five integrators are required for developing $d(UVW)$. Figure 19.59

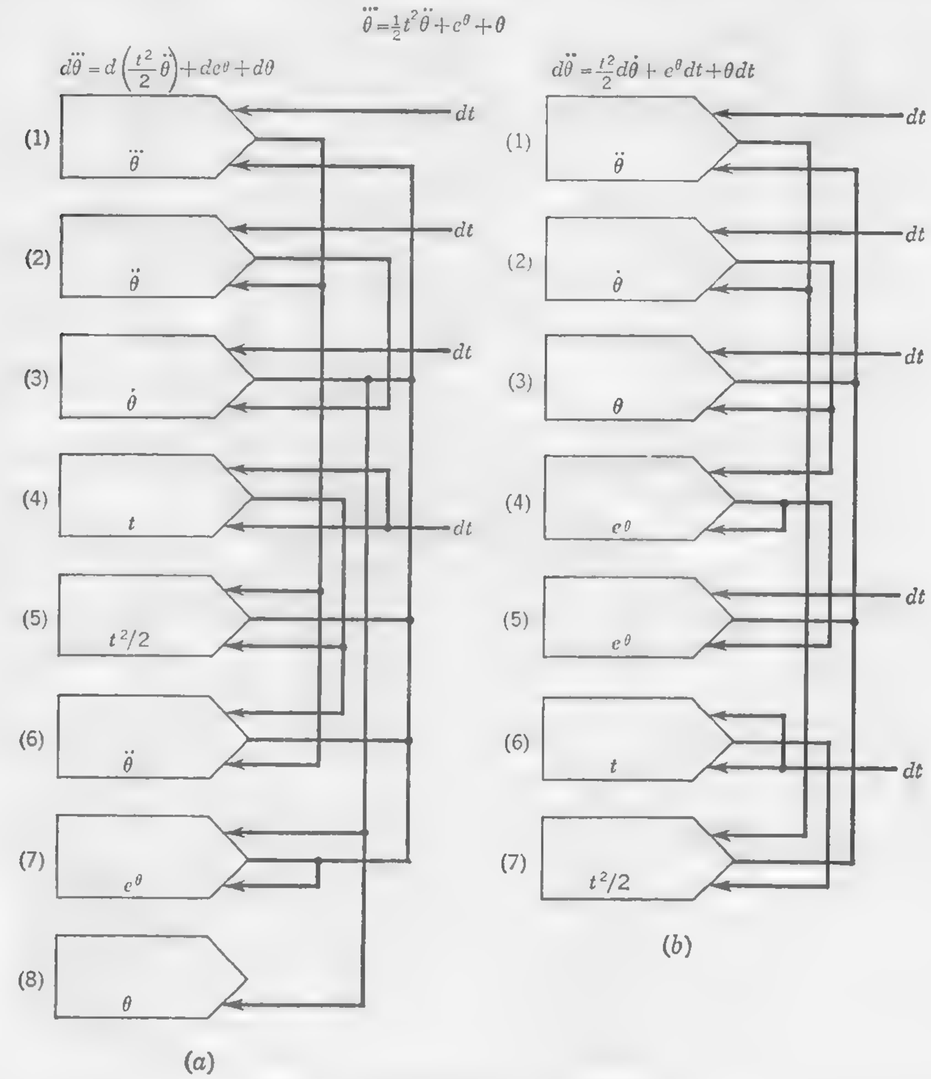


FIG. 19.59. Comparison of methods.

shows the application of both methods to a simple equation while Fig. 19.60 applies a procedure analogous to the method of Eq. (19.19) to the solution of the following set of simultaneous differential equations:

$$\frac{d^3W}{dt^3} = W\frac{dV}{dt} + V$$
$$\frac{d^3V}{dt^3} = W\frac{dW}{dt} + Vt$$

In some cases the application of either method will produce an integrand which becomes infinite at one or several points in a given problem. These cases require the use of a servo. In generating the solution of $y'' = (-1/x)y' - y$, for example,

the map shown in Fig. 19.61*a* cannot be used when x is small or zero. Figure 19.61*b* shows an alternate map that employs a servo to generate $1/x\ dy$.

The mapping of a problem may often be simplified by performing a substitution of variables. In Fig. 19.62, the independent variable of the computer is used to

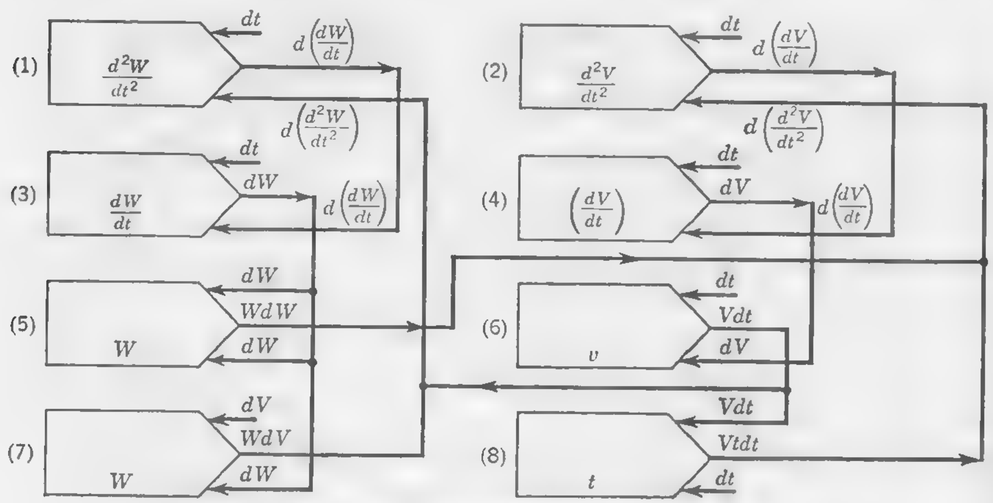


FIG. 19.60. Simultaneous equations.

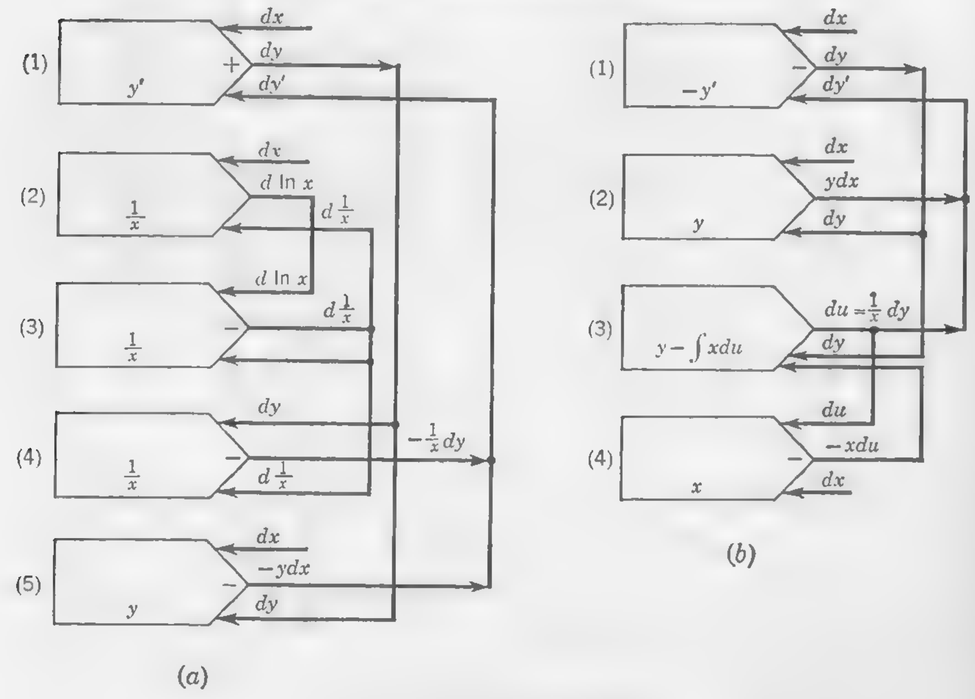


FIG. 19.61. Infinite integrands.

represent $d(\ln t)$ in order to solve the equation $y = \ln t$. It should be noted that this scheme is viable with the initial conditions $t = 1$, $y = 0$, even though y is infinite at this point.

For large problems the mapping notation discussed thus far becomes cumbersome. A more compact notation merely indicates the operation of each integrator. For example, the mapping shown in Fig. 19.61*a* may be symbolized as follows:

1. $dy = y'\,dx$
2. $d \ln x = (1/x)\,dx$
3. $d\,(1/x) = (-1/x)\,d \ln x$
4. $(-1/x)\,dy = (-1/x)\,dy$
5. $-y\,dx = -y\,dx$

In addition to specifying integrator interconnections, the mapping operation includes numbering the integrators and, in a digital differential analyzer with trapezoidal integration, assigning operational modes. For a discussion of these topics see page 19–42.

Scaling. Once a problem has been mapped, it must be scaled so as to yield numbers that the computer will process. The following exposition assumes a decimal radix,

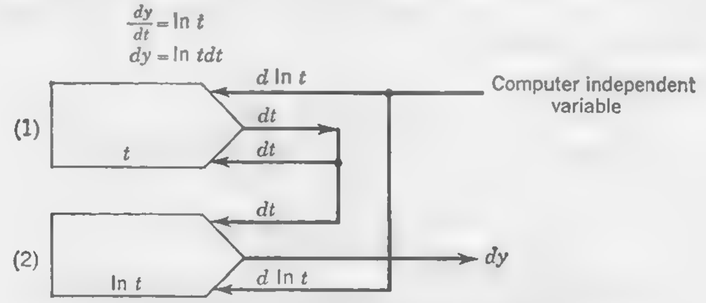


FIG. 19.62. Substitution of variables.

ternary intercommunication, and Y registers that take on the values $+1 > Y \geq -1$. Similar results are obtained if Y is restricted to integers.

The scaling relationships relate the computer variables (Y,dY,dX,dZ) and the problem variables (Y_p,dY_p,dX_p,dZ_p). One form these may take is

$$\begin{aligned} Y &= K_y Y_p \qquad dX = K_p\,dX_p \\ dY &= K_w\,dY_p \qquad dZ = K_z\,dZ_p \end{aligned} \tag{19.20}$$

where the K's are all positive and integral powers of the radix 10. The integrand scale factor K_y is such that

$$+1 < K_y Y_{p\,\max} \geq -1$$

while the incremental scale factors indicate the number of computer increments in one problem increment. Substituting the right-hand terms of Eqs. (19.20) into the basic integrator equation $dZ = Y\,dX$ gives

$$Kz\,dZ\,p = Ky\ Yp\ Kx\,dX\,p$$

which yields the basic scaling equation

$$Kz = Ky\,Kx$$

If, now, each K is expressed as a power of 10, then

$$\begin{aligned} 10^{Sz} &= 10^{Sy}10^{Sx} \\ Sz &= Sy + Sx \end{aligned}$$

The remaining problem is to determine the point at which the secondary inputs are to be added into the Y register. If the integrand register has the form such as in Fig. 19.63, a unit in the place immediately to the right of the computer decimal point has the value 10^{-Sy-1}. Since the value of each increment of dY is 10^{-Sw}, the number of places to the right of the decimal point N is given by

$$10^{-Sy-N} = 10^{-Sw}$$

so that

$$N = Sw - Sy$$

For example, if

$$\begin{aligned} Sy &= -2 \\ Sx &= 5 \\ Sw &= 4 \end{aligned}$$

then $Sz = -2 + 5 = 3$ and $N = 4 + 2 = 6$. The Y register therefore has the form shown in Fig. 19.64. Because the $Y + \Delta Y$ addition is initiated at a specific point in the integrator, all the secondary inputs must have the same scale factor. Furthermore, if several integrators supply the primary or secondary input to an integrator by means of a summing integrator, these also must be at the same scale. The scaling relationships that have been developed could be formalized into a set of equalities and inequalities which, for any given problem, have a large number of integral solutions. In practice, however, a solution is generally found by direct trial and error.

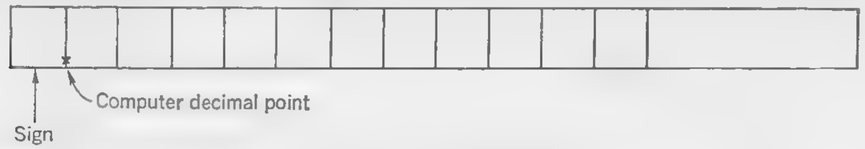


FIG. 19.63. Decimal points.

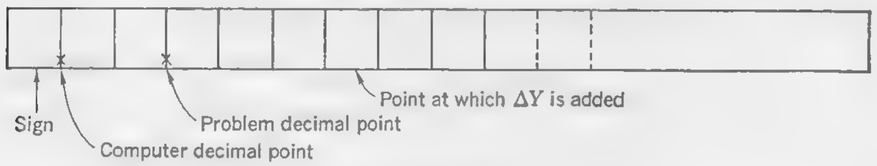


FIG. 19.64. Scaling.

In the event that integers are employed as output multipliers (see page 19–29), the basic scaling equation becomes

$$Kz = MKyKx$$

where M is the multiplier.

The scaling of a servo must be such that the rate at which error signals are produced is, on the average, less than a full rate. Moreover, if the delays in the output of a servo loop are to be kept at a minimum, the maximum magnitude of the error signal must be kept small. In terms of Fig. 19.65, the relationship between S_u, the scale of dU, and S_t, the scale of the computer independent variable, should be

$$S_u \geq \frac{S_t}{\left|\frac{dU}{dt}\right|_{\max}} \tag{19.21}$$

if the servo is to operate correctly. Furthermore, the scale of $dF(U)$ must be the same as the scale of $dV(t)$. Once these conditions have been met, dU may be employed elsewhere in the problem at a scale of S_u. In some instances these conditions cannot be met in every region of the solution. For example, if a servo is employed to generate $\sqrt{V}$, then $V(t) = U^2$ and $dU/dt = (1/2U)(dV/dt)$. As U approaches zero, dU/dt becomes very large so that S_u must approach zero with a given S_t. If, on the other hand, a given precision is to be maintained, S_t must be very large—which is generally impractical. However, if dV/dt is large, the resultant error is small and the solution quickly passes through the critical region. Furthermore, in many problems the lag introduced in U is not critical since only the value of U in the long run—after it has "caught up"—is of interest.

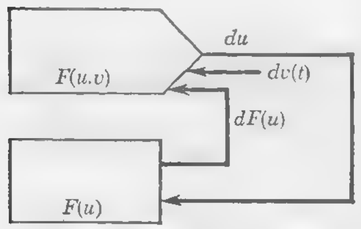


FIG. 19.65. Scaling of a servo.

The restriction on S_u may be removed by employing the program illustrated in Fig. 19.66 to generate the problem independent variable. This program assumes decision integrators of the type discussed on page 19–35 and operates to block increments of the problem independent variable whenever the error signal is one increment or more. For positive increments this is accomplished by blocking the output of integrator 1, for negative outputs by blocking the outputs of integrator 2. Since the output of integrator 2 is effectively in series with the output of integrator 1, either integrator can halt the generation of the solution until the outputs of integrator 3 again reduce the error to zero. By varying the constants held in integrators 1 and 2, the quiescent period may be reduced until the error becomes objectionable.

Like a servo, an adder is scaled so that the sum of the inputs is less than a full rate. This condition and the condition that the output scale factor S_u be equal to input scale factor lead to the scaling relationship

$$S_u \leq \frac{S_t}{\left|\frac{dU}{dt}\right|_{\max}}$$

where $U = V + W$ as shown in Fig. 19.33.

Figures 19.67 to 19.69 illustrate the scaling in decimal, with output multipliers 1, 2, and 5, and in binary of a single equation,

$$\ddot{v} + K\dot{v}(v^2 - 1) + v = 0$$

The maximum of $|\dot{v}|$ is 4.36; $|v|$ reaches 1.62 and K is 1.25. The scale of the independent variable is taken as 10^4 in the decimal cases and 2^{13} in the binary case; hence,

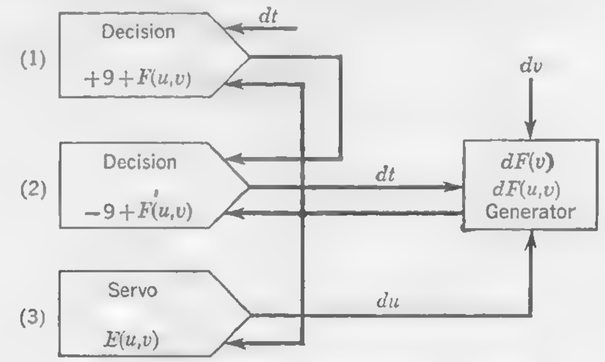
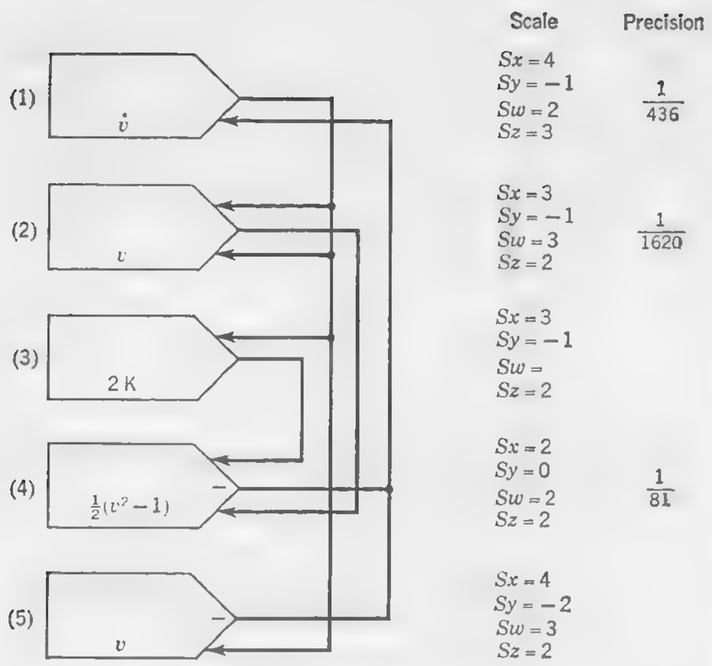


FIG. 19.66. Blocking independent variable.

in comparing the precisions of the three scaling schemes, the 22 per cent longer running time of the decimal computers with respect to the binary case should be kept in mind. In over-all accuracy, the decimal case is decidedly the worst while the remaining two yield comparable results. This example should not be construed as typical of the relative merits of the three schemes in general. For a discussion of these merits see page 19–29.

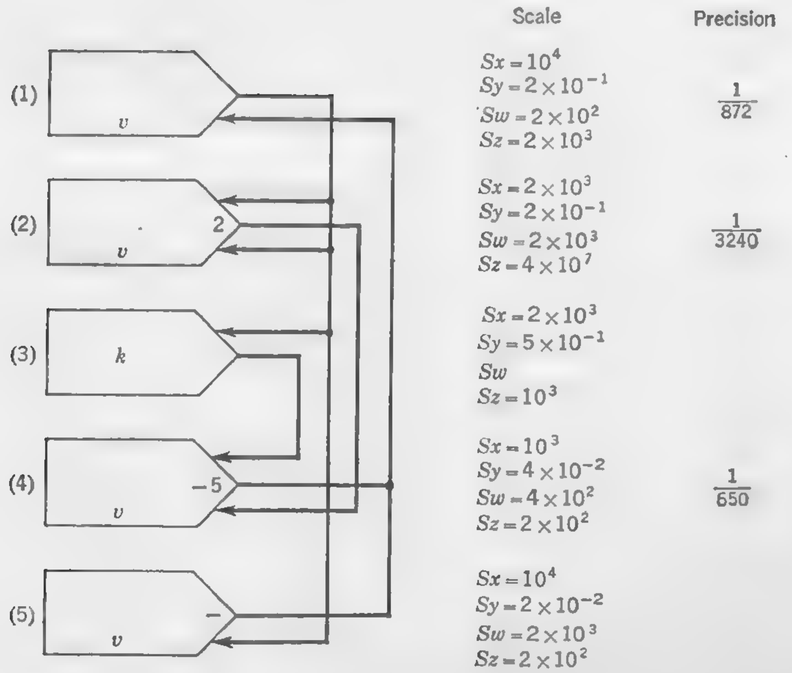
19.3.8. The Synthesis of General-purpose Computer and Digital Differential Analyzer Techniques. *Magnetic-drum Computers.* In many computer applications neither a general-purpose computer nor a digital differential analyzer provides an optimal computing device. The general-purpose computer, although theoretically capable of solving any mathematical problem that has a solution, is not efficient with respect to either programming or actual solution time in handling differential equations and certain related problems—particularly if the values of any of the variables must be generated at closely spaced points. A digital differential analyzer, on the other hand, has a restricted range of applications. A combination of the two computational techniques provides a flexible instrument for general computation, while in certain special-purpose applications it constitutes a solution that otherwise, because of restrictions on size, weight, cost, or computing time, could not be realized.

The combination of the two techniques is particularly applicable to a serial magnetic-drum computer (see Fig. 19.70) since the memory locations can serve simul-



Decimal scale factors

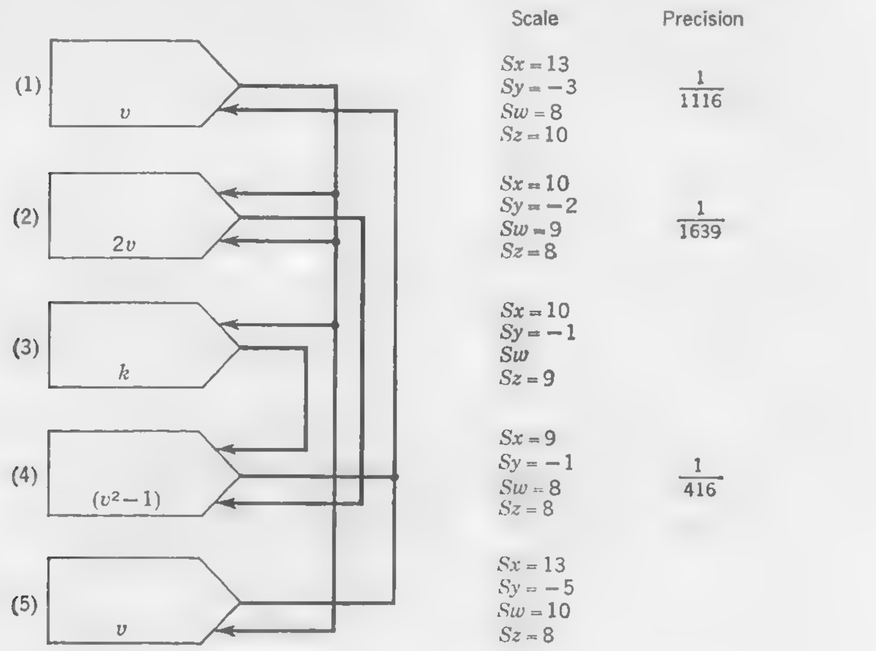
FIG. 19.67. Decimal scale factors.



Decimal scale factors with output multipliers 1, 2, 5

FIG. 19.68. Decimal scale factors with output multipliers 1, 2, 5.

taneously as integrators for the digital differential analyzer and addresses for the general-purpose computer. The difficulties usually associated with coupling two computers are thus eliminated since the general-purpose portion of the computer can, at all times, operate upon any of the information in the store. Furthermore, the



Binary scale factors

FIG. 19.69. Binary scale factors.

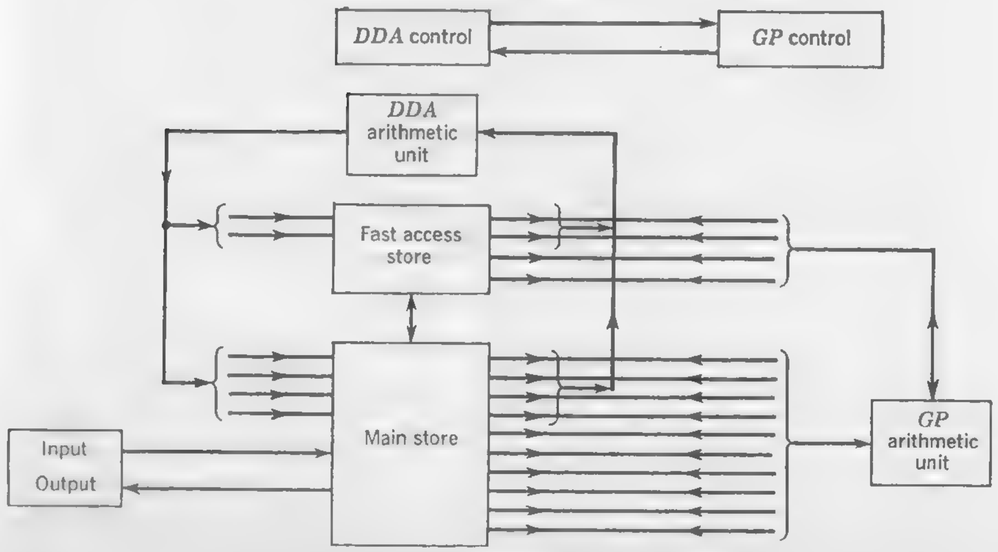


FIG. 19.70. DDA with a general-purpose computer.

relative location of all information remains fixed while the synchronizing problems do not arise because of the common clock.

The design of the general-purpose portion of a combination computer is unrestricted with the following exceptions: the memory must recirculate and commands must be

available for starting, stopping, and testing computation in the digital differential analyzer. A further requirement, which is normally not restrictive, is a fast-access store in which delay times are twenty-five to fifty times shorter than those in the main store. Several of the channels in this track are used for Z lines.

The design of the digital differential analyzer portion of the combination computer is a function not only of the requirements of the application but also of the configuration of the main and fast-access stores. For purposes of this exposition the following configuration is assumed:

32-bit word length
32 channels of 64 words in the main store
4 channels of 2 words in the fast-access store

Four of the main store channels can be used to provide for Y, R, K, and R_k registers while a fifth may serve to hold the start pulse. The inclusion of this latter channel simplifies the interaction of the general-purpose and the digital differential analyzer portions of the computer (see below) and reduces the round-off error (see page 19–46) at no added cost except a reduction in the storage capacity available to the general-purpose computer. In other respects the operation of the arithmetic channels is conventional.

The output distribution system, on the other hand, cannot be conventional unless additional equipment—short tracks with several reading heads—is provided. In lieu of this, two of the two-word channels may be employed as Z lines, one to indicate the existence of an output and the other the sign. During each word time, each of these lines receives a new integrator output and precesses one bit so that all 64 outputs may be held by the two lines. The output of a fixed integrator, which is artificially set to $+1$ each iteration, supplies computer time. The idle Y register of this integrator can serve to control output. Because two word times are required for all the integrator outputs to appear on the Z line, both the primary and the secondary input to an integrator must be collected during the *two* word times immediately preceding the processing of the integrator itself if universal integrator intercommunication is to be provided. As a consequence, the addressing system requires two sets of counters and registers: if one set is employed during I_{i-2} and I_{i-1} to collect inputs to I_i, another is needed during I_{i-1} to start collecting inputs for I_{i+1}. Separate address channels are required for the two sets of counters; hence four channels must be employed for addresses. Figure 19.71 illustrates both the conventional and this modified addressing system. If universal intercommunication is not necessary, only two channels and a single set of counters and registers may suffice.

A large variety of alternate address configurations are possible. Address and arithmetic information can, for example, occupy alternate words of the same channel. Or, if four-word short lines are employed in the above example, four sets of counters and registers and eight address channels may be employed to provide universal addressing. In this case, inputs are collected during four word times or 128 pulse times, only 64 of which contain the latest available information. The remaining 64 times contain older information which is normally not used. These and other complex addressing systems are feasible in a combination computer because the general-purpose portion of the computer can translate between a simple input language, in which the integrator outputs to be employed as inputs to a given integrator are indicated by number, and a complex Z-line configuration. Similarly, in a binary computer, decimal input data can be translated into binary and the output information translated back into decimal.

Many other operations can be performed by the general-purpose circuitry: input functions can be stored and entered into computation; iterative solutions involving the successive variation of initial conditions can be controlled and directed; difference tables can be generated for either input or output; initial conditions can be computed from raw input data; portions of a problem that do not lend themselves to solution by a digital differential analyzer can be computed in parallel with the digital differential analyzer solution; etc. One particularly significant function that the general-purpose portion of the computer can perform is the rescaling of a problem as a function of the solution. The degree to which this is made automatic is dependent upon the application and the memory capacity that remains once other functions have been provided. The rescaling operation can be programmed specifically for each problem or in various degrees of generality. Moreover, it may operate only in the event that a variable exceeds a specified value or, more generally, it may serve to optimize the efficiency of computation by repositioning variables and varying scale factors. The inclusion of a separate channel for the start pulse facilitates the implementation of any of these rescaling processes.

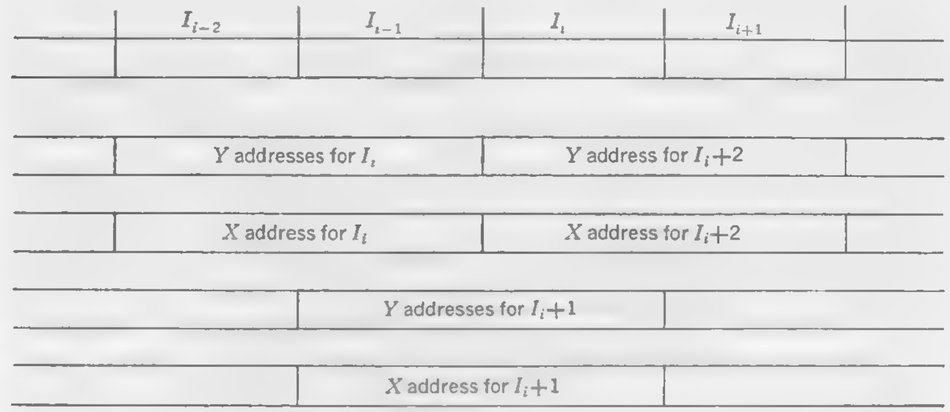


FIG. 19.71. Conventional and modified addressing.

Other Types of Synthesized Computers. The computer discussed in the last section represents a combination of digital differential analyzer and general-purpose techniques in which the two parts of the computer are coordinate and closely interrelated. In other applications, one of the parts may be subordinated to the other or the two techniques may be embodied in separate and independent computers which are coupled for the solution of special problems.

The inclusion of a small general-purpose computer for auxiliary operations, such as the processing of input and output data and the conversion of numbers, may be warranted by the cost of a large parallel digital differential analyzer. This auxiliary computer may have fixed or variable programming which operates under the control of the variables within the digital differential analyzer. On the other hand, in some real-time applications a small digital differential analyzer may be used in conjunction with a general-purpose computer to generate certain continuous control variables or other functions.

The most important application of the combination of the two techniques appears to be in large-scale real-time simulations of complex systems—such as the simulation of an intercontinental ballistic missile. For real-time calculations of this magnitude present-day general-purpose computers are several orders too slow while analog computers lack the required accuracy. One approach to this problem is a combination of a general-purpose and an analog computer, since analog accuracy is sufficient for certain computations. In this case, however, a large number of analog-to-digital and digital-to-analog conversions are required in addition to the analog-to-analog conversions that are necessary if the simulation is to include portions of the physical system being simulated. Replacing the analog computer by a parallel digital differential analyzer reduces the number of these conversions, eliminating the former and replacing the latter by digital transducers. Furthermore, because of its greater accuracy, a digital differential analyzer can generally handle a larger portion of the problem, thereby allowing the general-purpose computer to operate upon equations that represent a less simplified model of the equipment being simulated.

A digital differential analyzer may also operate in conjunction with an analog computer in order to provide digital accuracy at those critical points in analog computation which require the use of nonlinear elements, such as servos and function generators. In some cases digital equipment can also serve to buffer information between analog equipment and portions of a system under investigation. In these cases digital transducers can be used to replace analog-to-analog converters.

References

1. Amble, O.: On a Principle of Connexion for Bush Integrators, *J. Sci. Instr.*: 284–287 (December, 1946).
2. Archer, A. I.: Vendac—A Venn Diagram Analogue Computer, *S. African J. Sci.*, **47**(5): 133–140 (December, 1950).
3. Berkeley, E. C.: *Giant Brains*, chap. 9, John Wiley & Sons, Inc., New York, 1949.
4. Blanyer, C. G., and H. Mori: Analog, Digital, and Combined Analog-digital Computers for Real-time Simulation, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 104–110.
5. Boole, George: *Mathematical Analysis of Logic*, G. Bell & Sons, Ltd., London, 1847.
6. Boole, George: *An Investigation of the Laws of Thought*, 1854, Dover Publications, Inc., New York, 1951.
7. Bowden, B. V. (Ed.): *Faster Than Thought*, Sir Isaac Pitman & Sons, Ltd., London, 1953.
8. Braun, Edward L.: Design Features of Current Digital Differential Analyzers, *IRE Convention Record*, part 4, New York, 1954, pp. 87–97.
9. Braymer, Noel B.: A Computer for Flaw Plotting, *IRE Trans. on Electronic Computers*, **EC-1**:73–75, (December, 1952).
10. Burbeck, D. W., E. E. Bolles, W. E. Frady, and E. M. Grabbe: The DIGITAC Airborne Control System, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 38–44.
11. Burks, A. W., D. W. Warren, and J. B. Wright: An Analysis of a Logical Machine Using Parenthesis-free Notation, *Mathematical Tables and Other Aids to Computation*, **8**(46): 53–57 (April, 1954).
12. Bush, V.: Differential Analyzer, *J. Franklin Inst.*, **212**(4): 447–488 (October, 1931).
13. Bush, V.: Differential Analyzer, *Mech. Eng.*, **54**(1): 56–57 (January, 1932).
14. Clark, W. A., and B. G. Farley: Generalization of Pattern Recognition in a Self Organizing System, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 86–91.
15. Clos, C.: A Study of Non-blocking Switching Networks, *Bell System Tech. J.*, **32**(2): 406–424 (March, 1953).
16. Courant, R., aud D. Hilbert: *Die Methoden der mathematischen Physik*, vol. II, Springer-Verlag, Berlin, 1937.
17. Cox, B., and J. Goldberg: A Magnetic-drum Sorting System, *IRE Convention Record*, part 4, New York, 1956, pp. 101–104.
18. Crank, J. M.: *The Differential Analyzer*, Longmans, Green & Co., Ltd., London, 1947.
19. Dinneen, G. P.: Programming Pattern Recognition, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 94–100.
20. Donan, J. F.: The Serial-memory Digital Differential Analyzer, *Mathematical Tables and Other Aids to Computation*, **6**(38): 102–112 (April, 1952).

21. Dunn, W. H., C. Eldert, and P. V. Levonian: A Digital Computer for Use in an Operational Flight Trainer, *IRE Trans. on Electronic Computers*, **EC-4**(2): 55–63 (June, 1955).
22. Forbes, G. F.: *Digital Differential Analyzers*, published by the author, Pacoima, Calif., 1956.
23. Fort, T.: *Finite Differences and Difference Equations in the Real Domain*, Oxford University Press, Oxford, 1948.
24. Frankel, S.: Convergence Rates of Iterative Treatments of Partial Differential Equations, *Mathematical Tables and Other Aids to Computation*, **4**(30): 65–75 (April, 1950).
25. Gordon, B. M., and R. N. Nicola: Special Purpose Digital Data-processing Computers, *Proc. Assoc. Computing Machinery*, Pittsburgh, May 2–3, 1952, pp. 33–45.
26. Gray, H. J., Jr.: Numerical Methods in Digital Real-time Simulation, *Quart. Appl. Math.*, **12**(2): 133–140 (July, 1954).
27. Hartree, D. R.: *Numerical Analysis*, Oxford University Press, Oxford, 1952.
28. Hartree, D. R.: *Calculating Instruments and Machines*, University of Illinois Press, Urbana, Ill., 1949.
29. Hill, John L.: Design Features of Remington Rand Speed Tally, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 155–162.
30. Jevons, W. S.: On Mechanical Performance of Inference, *Nature*, **1**:343–344 (Jan. 27, 1870).
31. Jevons, W. S.: On a Logical Abacus, *Proc. Manchester Lit. Phil. Soc.*, **5**:161–165 (Apr. 3, 1866).
32. Johnson, R. R.: An Electronic Digital Polynomial Root Extractor, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 119–123.
33. Ketchledge, R. W.: An Introduction to the Bell System's First Electronic Switching Office, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 204–208.
34. Larson, H. T., and A. Vazsonyi: Data Processor Requirements in Production and Inventory Control, *Proc. Western Joint Computer Conf.*, Mar. 1–3, 1955, pp. 48–61.
35. Leonard, Eugene: Characteristics of a Logistics Computer, *Proc. WESCON Computer Sessions*, Los Angeles, Calif., Aug. 25–27, 1954, pp. 77–85.
36. Leondes, C., and M. Rubinoff: DINA, A Digital Analyser for Laplace, Poisson, Diffusion, and Wave Equations, *Trans. AIEE, Communications and Electronics*, **71** (part 1): 303–309 (November, 1952).
37. Lesh, F.: Methods of Simulating a Differential Analyzer on a Digital Computer, *J. Assoc. Computing Machinery*, **5**(3): 281–288 (July, 1958).
38. Lukasiewicz, Jan: *Aristotle's Syllogistic from the Standpoint of Modern Formal Logic*, Oxford, 1951, p. 78.
39. Malthaner, W. A., and H. E. Vaughan: An Automatic Telephone System Employing Magnetic Drum Memory, *Proc. IRE*, **41**(10): 1341–1347 (October, 1953).
40. Malthaner, W. A., and H. E. Vaughan: An Experimental Electronically Controlled Automatic Switching System, *Bell System Tech. J.*, **31**(3): 443–468 (May, 1952).
41. Malthaner, W. A., and H. E. Vaughan: Control Features of a Magnetic-drum Telephone Office, *IRE Trans. on Electronic Computers*, **EC-4**(1): 21–26 (March, 1955).
42. Marquand, A.: On Logical Diagrams for *n* Terms, *Phil. Mag.*, **12**:266–270 (1881).
43. Marquand, A.: A New Logical Machine, *Proc. Am. Acad. Arts Sci.*, **21**:303–307, (1885–1886).
44. Mays, W., and D. P. Henry: Exhibition of the Work of W. Stanley Jevons, *Nature*, **170**: 696–697 (Oct. 25, 1952).
45. McCallum, D., and J. Smith: Mechanized Reasoning Logical Computers and Their Design, *Electronic Eng.*, **23**(278): 126–133 (April, 1951).
46. Meissner, Loren P.: Real-time Digital Differential Analyzer ("DART"), *Trends in Computers: Automatic Control and Data Processing*, S-59, *Proc. Western Joint Computer Conf.*, Los Angeles, Feb. 11–12, 1954, pp. 134–139.
47. Mendelson, M. J.: The Decimal Digital Differential Analyzer, *Aeronaut. Eng. Rev.*, **13**(2): 42–54 (February, 1954).
48. Michel, J. G. L.: Extensions in Differential Analyser Technique, *J. Sci. Instr.*: 357–361 (October, 1948).
49. Milne-Thompson, L. M.: *The Calculus of Finite Differences*, The Macmillan Company, New York, 1933.
50. Nelson, Eldred: A Digital Computer for Airborne Control Systems, *IRE Trans. on Electronic Computers*, **EC-1**: 2–5 (December, 1952).
51. Oliver, B. M., J. R. Pierce, and C. E. Shannon: The Philosophy of PCM, *Proc. IRE*, **36**(11): 1324–1331 (November, 1948).

52. Palevsky, M.: An Approach to Digital Simulation, *Proc. Natl. Simulation Conference*, Dallas, January, 1956, pp. 18.1–18.4.
53. Palevsky, M.: The Design of the Bendix Digital Differential Analyzer, *Proc. IRE*, **41** (10): 1352–1356 (October, 1953).
54. Richards, R. K.: *Arithmetic Operations in Digital Computers*, D. Van Nostrand Company, Inc., Princeton, N.J., 1955.
55. Selfridge, O. G.: Pattern Recognition and Modern Computers, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 91–93.
56. Selfridge, R. G.: Coding a General-purpose Digital Computer to Operate as a Differential Analyzer, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 82–84.
57. Shannon, C. E.: A Symbolic Analysis of Relay and Switching Circuits, *Trans. AIEE*, **57**:713–723 (1938).
58. Shannon, C. E.: Mathematical Theory of the Differential Analyser, *J. Math. Phys.*, **XX**(4): 337–354 (December, 1941).
59. Shannon, C. E., and E. F. Moore: Machine Aid for Switching Circuit Design, *Proc. IRE*, **41**(10): 1348–1351 (October, 1953).
60. Shull, J. R.: An Automatic Cruise Control Computer for Long Range Aircraft, *IRE Trans. on Electronic Computers*, **EC-1**: 47–51 (December, 1952).
61. Singleton, H. E.: A Digital Electronic Correlator, *Proc. IRE*, **38**(12): 1422–1428 (December, 1950).
62. Skabelund, D. E.: The Numerical Process of a Binary Differential Analyser, *University of Utah Rept.*, August, 1953.
63. Sommerfeld, A.: *Partial Differential Equations in Physics*, Academic Press, Inc., New York, 1949.
64. Sprague, R. E.: Fundamental Concepts of the Digital Differential Analyzer Method of Computation, *Mathematical Tables and Other Aids to Computation*, **6**(37): 41–49 (January, 1952).
65. Sprague, R. E.: CRC-105 Computer, *Aero Digest*, **67**(2): 48–55 (August, 1953).
66. Stone, J. J.: The USAF-Fairchild Specialized Digital Computer, *Mathematical Tables and Other Aids to Computation*, **7**(41): 35–37 (January, 1953).
67. Tucker, C.: A Digital Electronic Musical Instrument, M.S. Thesis, University of California at Los Angeles (January, 1956).
68. Venn, J.: On the Diagrammatic and Mechanical Representation of Propositions and Reasonings, *Phil. Mag.*, **10**:1–18 (July, 1880).
69. Walker, J. P., Jr.: A Special Purpose Digital Computer, *Mathematical Tables and Other Aids to Computation*, **7**(3): 190–195 (July, 1953).
70. Weiss, Eric: Applications of CRC-105 Decimal Digital Differential Analyzer, *IRE Trans. on Electronic Computers*, **EC-1**: 19–24 (December, 1952).
71. Whittaker, E. T., and C. Robinson: *Calculus of Observations*, Blackie & Son, Ltd., Glasgow, 1944.
72. Wiener, N.: *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, John Wiley & Sons, Inc., New York, 1949.

Section 20

GENERAL-PURPOSE COMPUTERS

By

ERNEST G. ANDREWS, *Manager, Preliminary Design, Sanders Associates, Nashua, N.H.*

WILLIAM R. ARSENAULT, *Vice-President, FMA, Inc., El Segundo, Calif.*

HARRY D. HUSKEY, *Professor of Electrical Engineering and Mathematics, University of California, Berkeley, Calif.*

CONTENTS

20.1. General-purpose Digital Computers—Relay Types ... 20–2
20.1.1. Historical Aspects ... 20–2
20.1.2. Characteristics ... 20–3
20.1.3. Input Devices ... 20–5
20.1.4. Output Devices ... 20–5
20.1.5. Summary ... 20–5
20.2. Methods of Classifying Stored-program Computers ... 20–5
20.2.1. Grand Central Station vs. Source Destination ... 20–5
20.2.2. Memory Hardware ... 20–6
20.2.3. Number Representation ... 20–6
20.2.4. Fixed-point vs. Floating-point Arithmetic ... 20–6
20.2.5. Decimal vs. Binary ... 20–7
20.2.6. Command Sequencing ... 20–7
20.3. Special Characteristics of Magnetic-drum Computers ... 20–7
20.3.1. Buffer Memories ... 20–7
20.3.2. Circulating Registers ... 20–8
20.3.3. Loop Realization and Machine Timing ... 20–8
20.3.4. Shift Operations ... 20–9
20.4. Some General-purpose Drum Computers and Their Characteristics ... 20–10
20.4.1. The IBM 650 ... 20–10
Memory ... 20–10
Word Structure ... 20–10
Logic ... 20–10
Command Execution ... 20–11
Equipment ... 20–12
Auxiliary Equipment ... 20–12
20.4.2. The Royal-McBee LGP-30 20–12
Memory ... 20–12
Word Structure ... 20–12
Order Code ... 20–12
Logic ... 20–12
Multiplication ... 20–14
Machine Timing ... 20–14
Address Selection ... 20–15
Sector Interlacing ... 20–15
Equipment ... 20–15
Auxiliary Equipment ... 20–15
20.4.3. The Bendix G-15 Computer ... 20–15
Memory ... 20–15
Inverting Gates ... 20–15
Arithmetic Operations ... 20–17
Input and Output ... 20–17
The Command Cycle ... 20–17
Precession ... 20–17
The Digital Differential Analyzer Attachment ... 20–19
20.5. Large-scale Digital Computers ... 20–19
20.5.1. Memory Hierarchy ... 20–19
High-speed Memory ... 20–19
Medium-speed Memory ... 20–19
Slow-speed Memory ... 20–19
General Memory Organization ... 20–19
20.5.2. Arithmetic System ... 20–19
20.5.3. Control System ... 20–20
20.6. A Simple One-address Computer ... 20–20
20.7. Sequence Scheme for Arithmetic Operation ... 20–23

20.8. Transfer-of-control Operations 20–24
20.9. A Simple Two-address Computer 20–24
20.10. Multiple-address Computers 20–25
20.11. Index Registers 20–27
20.12. The Address Modifier 20–29
20.13. A Computer with Page Memory and Interrupt Logic 20–30

20.1. GENERAL-PURPOSE DIGITAL COMPUTERS—RELAY TYPES*

20.1.1. Historical Aspects. The modern digital computers are electronic just as are most other modern data-handling and control systems. However, they were all first conceived as mechanical-type systems. In between these extremes of implementation came the electromechanical types—more commonly called relay types

Table 20.1. Data on Relay-type Digital Computers

Item No.	Popular name of computer	Developed by	Approx No. of relays*	Service date	Now in service?	Number built
1	Type 600 multiplying punch	IBM	200 (60)†	1930	Many	2,500
2	Complex-number computer (model 1)	Bell	450	January, 1940	No	1
3	Interpolator (model 2)	Bell	450	September, 1943	Yes	1
4	Ballistic-data computer (model 3)	Bell	1,500	June, 1944	Yes	1
5	Harvard (Mark I)	IBM	3,304 (2,200)†	August, 1944	Yes	1
6	High-speed relay calculators	IBM	2,900	December, 1944	No	5
7	Ballistic-data computer (model 4)	Bell	1,550	March, 1945	Yes	1
8	General purpose (model 5)	Bell	8,900	December, 1946	Yes	2
9	Harvard Mark II	Harvard	13,000	May, 1947	Yes	1
10	Selective-sequence electronic calculator	IBM	21,400 (12,500)‡	January, 1948	No	1
11	Harvard Mark III	Harvard	1,250 (4,500)‡	September, 1949	Yes	1
12	Network computer (model 6)	Bell	4,700 (84)‡	September, 1949	Yes	1
13	Bark computer	Swedish Telephone Company	5,200	1950	No	1

* Approximate number of relays as originally built. Several computers now have more relays than indicated.
† Numbers in parentheses indicate number of Hollerith counters used.
‡ Numbers in parentheses indicate number of vacuum tubes used.

because this apparatus component performed the more dramatic part of the computing operations. The relay-type digital computer dominated the field for only a few years, primarily because this kind of digital-computer development was initiated on a large scale just before large electronic systems were deemed to be feasible.

The pioneering of the digital computer was done in the 1930s. Among those prominent in the field were Lake,* Aiken,† and Stibitz.‡ Each was identified, in the order named, with the three organizations principally responsible for the design and the construction of relay computers, namely, the International Business Machines Corporation, the Harvard Computation Laboratory, and the Bell Telephone Laboratories.

* By Ernest G. Andrews.

The whole era of the relay-type digital computer was so short that only a few were built. The data on those which have no security restrictions are covered in Table 20.1. In the transition from relay types to electronic types there were a few computers which had substantial parts of both designs, but as the use of the electronic techniques implied a much faster computer these machines have been generally regarded as electronic computers.

The use of relays in designing the first large digital computer in the 1930s came about naturally. The circuitry then employed in automatic data-handling systems, signaling systems, and automatic telephone switching had the basic requisites for digital-computer design. The relay computer has many virtues compared with its predecessor, the mechanical type: (1) a single signal operating a relay can control many paths, up to 24 in conventional high-speed types and many more paths for special types; (2) the relay has one moving part, the relay armature (neglecting the flexing of contact springs); (3) reduced maintenance; (4) self-checking operations are more easily incorporated; and (5) lower initial cost. In addition, the relay brought about faster operation but the degree of improvement was not great compared with that of electronic computers.

The early relay computers developed by IBM also employed another type of electromechanical device called the Hollerith counter. From a functional viewpoint this counter could be described as a relay having 10 discrete states, and it obviously was therefore well adapted to handling decimal numbers. This counter is analogous to the numeral wheel on the mechanical calculating machines because storage was effected by turning the wheel to the unique position corresponding to the decimal digit to be stored.

Many of the principles incorporated in modern electronic computers were pioneered in the relay-type designs: (1) variable computing program (provided by punched tapes), (2) the use of the binary code for computing, (3) floating-decimal-point operation, (4) the multiple-address instruction code, (5) error detection, (6) conditional transfer of control operations, and (7) effective use of the subroutine concept.

20.1.2. Characteristics. *Code.* Several forms of decimal-number codes were employed among which were (1) the plus-3 binary coded decimal, (2) the 8421 code (true binary), (3) the 2*421 (gap 6 code), (4) the biquinary code (a 7-unit code), and (5) various forms of the 5-unit code. All these were used in the arithmetic units. In addition, still other codes were employed in input and output devices.

One computer used an all-binary code.

Negative numbers were handled either as complements or as the absolute number with a sign indicator.

Storage. High-speed storage of numbers was handled by relays, four to seven being required per decimal digit (depending upon the digital code employed) plus the necessary gating and control relays. The cost of a number register was therefore so

* Claire D. Lake is identified with the development of the Harvard Mark I Computer. As indicated in the *New York Herald Tribune* of Aug. 7, 1944, this computer was placed in operation and publicly demonstrated on August, 1944, at which time the IBM Corp. gave it to Harvard University.

† Howard Aiken is identified with attempting to promote the idea of automatic sequence control in April, 1937. At that time he outlined the general design of a computing machine primarily for use in the field of mathematics, science, and sociology. (See pages 25 and 26 of *Proc. ACM* meeting held at Pittsburgh, Pa., on May 2 and 3, 1952.) Aiken subsequently assumed an active role in the development of the Harvard Mark I.

‡ George R. Stibitz is identified with the Bell Telephone Laboratories complex-number computer. It was placed in operation on Jan. 8, 1940, and publicly demonstrated on a remote-control-operation basis at Dartmouth University on Sept. 11, 1940.

high that a great deal of program-design consideration had to be given to keeping the number of registers as low as possible.

While the relay was an inefficient device as a storage medium for recording a single binary element of a code, extra relay contacts were readily obtainable. These were used to provide grid networks for various kinds of special controls not obtainable in simpler types of binary storage. Some examples are (1) self-checking features, and their accompanying alarm features; (2) certain numerical criteria could be indicated such as whether a number is zero, finite, or infinity (i.e., beyond the machine's number range), this information being used to cope with indeterminate arithmetic situations; and (3) code conversions such as conversion from the code of the arithmetic unit to the code of the output device.

The slow-speed storage was handled principally by paper tapes. In some cases punched cards were used.

Tabular data, either transcendental functions or empirical data involving one or more variables, were usually stored on punched tapes. For problems where random access to the tabular data was required, a disproportionate amount of access time was involved because a tape is a unidimensional device. To reduce this access time, it was customary to enter the table at high tape-hunting speed with coarse values of the argument and also to use either or both of these techniques: (1) provide several tapes (and their associated tape-reading equipment) for the table, and (2) store the data as coefficients of a second- or higher-order polynomial.

Wired-in function tables were used infrequently.

Storage of computing instructions was on paper tape. In general this type of storage was compatible with the over-all operating time of the control system, and consequently high-speed storage of instructions was not needed.

Control. It was generally recognized that relay computers were properly labeled as being slow. Numerous ingenious devices were therefore employed to speed up the computer. Various combinations of the following principles were used: (1) several arithmetic units were employed in a single computer; (2) the multiple-address instruction code was commonly used; (3) square rooters were provided; (4) parallel operation was almost universally employed; and (5) provision was made for the operation of two complete computers together when operating personnel required problem solutions in minimum time.

Arithmetic Unit. All arithmetic units operated in the parallel mode; as previously stated serial operation was too slow to be considered.

Addition was effected by providing two relay-contact networks on each of the groups of relays employed for each decimal digit position of the two operands. One of these was energized when the carry-out from the decimal digit on the right was 0 and the second network when the carry-out was 1.

Subtraction was effected by adding the complement of one of the operands. The decimal codes employed usually made it possible to obtain the complement by simple switching. When the arithmetic unit was required to produce the complement, an alternative gating relay to the adder was used. The normally used gating relay would transfer the operand to the adder without change; the alternative gating relay would transpose the wiring so that the complement was transferred to the adder. With this kind of circuitry, complementing was obtained without any added operating time.

Multiplication was effected by several methods. One computer, as a preliminary operation, obtained the various multiples of the multiplicand and selected these as required by the multiplier and with the proper shifting. Other computers used the repeated-addition technique; i.e., the multiplicand was added without shifting as many times as indicated by a multiplier digit. Other computers used a so-called "short-cut" repeated addition. This was a scheme which took advantage of the fact that multiplication with a multiplier digit from 1 to 4 could be accomplished with 1 to 4 normal additions; and with a multiplier digit from 5 to 9, multiplication could be accomplished with 5 to 1 subtractions in the next multiplier digit position. An appreciable reduction in multiplying time was thus obtained at the expense of more circuit complexity. Other computers performed multiplication using multiplication-table techniques.

Division was likewise effected by several methods. One computer referred to a table of reciprocals and thereby performed division by multiplying the minuend by the reciprocal of the divisor. More commonly, computers performed division by repeated subtraction. Another computer effected a reduction in computing time by first subtracting the divisor multiplied by 5; if the result had one sign, the quotient digit was known to be in the range of 0 to 4, the particular quotient value being obtained by repeated subtraction. If the result of the subtraction by the divisor multiplied by 5 was the other sign, the partial result was retained and the quotient digit was known to be 5 to 9, the particular quotient value being obtained by repeated subtraction of the divisor from the retained partial result.

Square rooting was obtained by subtracting trial root divisors made up from successive odd numbers, a scheme commonly used by operators of hand computing machines.

20.1.3. Input Devices. Punched-tape- and punched-card-type readers were most commonly used for both problem data and program data. Despite their slow speed, they were reasonably compatible with the over-all operation time of relay-type machines. For the two computers using electronic techniques (items 9 and 10 in Table 20.1), it was necessary to provide buffer storage facilities between the paper-tape inputs and the computer in order to capture more fully the value of the higher speed of computing afforded by electronic circuitry.

20.1.4. Output Devices. Electrically controlled typewriters or teletype printers were often used without any buffering. Punched cards, punched tapes, and tabulators were also commonly used. One computer (item 11 of Table 20.1) used magnetic-tape outputs.

20.1.5. Summary. When the technological advances immediately before World War II demanded automatic digital computing, the relay-type computer was the answer. It provided all the functional needs of mass computations; in fact, there are probably very few computing functions in the more modern computers which were not pioneered, sometimes crudely, in the relay-type digital computer.

20.2. METHODS OF CLASSIFYING STORED-PROGRAM COMPUTERS

Stored-program computers may all be classified as serial, where the serial refers to binary bits, decimal digits, words, or numbers, and perhaps to complete formula statements. A computer which is serial digit by digit transfers the various bits which represent the digits on parallel lines, whereas a computer which is serial by word may transfer all the bits that represent the word on a number of lines. For example, a 40-binary bit number would be transferred on 40 different channels.

Computers which are serial bit by bit may be constructed using acoustic delay line memory, or magnetic drum memory, or even a Williams tube memory. The first computer constructed using electrostatic storage of information was the computer at Manchester University where the information was processed in a strictly serial bit-by-bit form.

Computers which are serial by digit have been made using magnetic drum memories with the four binary bits representing individual decimal digits being transmitted on four different channels.

Computers which are serial by word transfer the various binary bits representing the word on different channels at the same time. Such computers have been referred to as parallel computers. The type of memory used in such parallel computers has been the Williams tube memory and more recently magnetic core memories.

For a given pulse-repetition rate the speeds of the various kinds of computers go up in proportion to the number of bits which are transferred in parallel. Thus the serial by word, or parallel computer, is the fastest of the group. Also, because of the multiple channels required, it makes use of much more equipment than the serial by digit or serial by binary bit computers.

20.2.1. Grand Central Station vs. Source Destination. Another method of comparing digital computers is to consider the general flow of information during the processing sequence. In many computers there is a central register, perhaps a memory buffer register, through which all the information is transferred, both the numbers which are

being operated upon and the commands which control the process. In contrast to this there is a design in which information may be transferred from any location in the memory system to almost any other location. In this type of system, sometimes called the source-destination system, there is no single register through which the information passes. However, there is usually a central bus over which all the information passes.

Generally, the logical structure and the command system of the Grand Central Station type of computer are simpler than those for the source-destination type. By the same token, the flexibility of the Grand Central Station system is less than that of the source-destination system. Note that whereas there is no means of extending the flexibility of the Grand Central Station system it is possible to establish program methods in the source-destination system which reduce the flexibility and make it behave as if it were a Grand Central Station system.

20.2.2. Memory Hardware. Another method of classifying computers is to consider the kind of hardware which is used to construct the memory portion. The types of memory in most prominent use are magnetic drum and magnetic cores. In the initial development of computers both Williams tube and acoustic delay lines were used, but these are less popular in the more recent developments. The distinction with respect to hardware correlates with respect to speed, and to a considerable extent with the distinction with respect to serial by bit or digit, and serial by word. Thus a magnetic drum computer tends to be serial by bit or by digit in construction, whereas the magnetic core computer tends to be serial by word.

20.2.3. Number Representation. The numbers that are processed in a digital computer may be represented in sign and absolute-value form, or negative numbers may be represented in complementary form. The complements may be with respect to either 9's or the appropriate power of 10 in a decimal computer, or with respect to 1's or the appropriate power of 2 in a binary computer. The various representations may hold whether the numbers are in fixed-point form or in floating-point form. The appearance of numbers in complementary form may be limited to the arithmetic circuits, or numbers stored in the memory may also be in complementary form. The conversion to sign and absolute-value or true form may occur only in the input and output equipment.

20.2.4. Fixed-point vs. Floating-point Arithmetic. Another distinguishing characteristic in the arithmetic systems of digital computers is whether the system operates using fixed-point numbers or floating-point numbers. Fixed-point numbers may be numbers which lie between $+1$ and -1, whereas floating-point numbers may range from -10^{50} to $+10^{50}$. This more extensive range makes it possible to program most scientific or engineering calculations, and not worry about variables in the computation overflowing during the process. On the other hand, in the fixed-point computation, the programmer must arrange that at each step of the way there is no danger that the variables developed in the computation will exceed 1 in absolute value. This requires, for example, that before each division takes place the programmer makes certain that the numerator is going to be less than the denominator in absolute value.

In a fixed-point computer it is sometimes desirable to compute with the numbers as if they were integers. This requires a different kind of multiplication than is used in the fractional case. In many computers the multiplication command has the effect of multiplying two numbers which may occupy word positions in the computer so as to produce a two-word answer, a more significant word and a less significant word. The distinction of whether one is multiplying fractions or integers occurs when the storage operation takes place. If the least significant half is stored this may be the integral answer to a product of integers, whereas if the most significant half is stored this is the fractional answer to the product of two fractions. This also makes it convenient to program double-precision operations in the same fixed-point arithmetic system.

In floating-point arithmetic systems there is a further choice as to whether results are normalized after each arithmetic operation. If the results are normalized, there is a tendency to produce more accurate results in larger computations. However, if normalization takes place nonsignificant digits may be brought in on the right-hand

end of numbers appearing in the computation; so that there is no indication whether the results of lengthy computations have significance or not. This has led to the use of nonnormalized systems, wherein there is no shifting of operands left, except in the case of multiplication and division operations. In this case there is more information about the significance of the answers. On the other hand, the distributions are such that perhaps there is less accuracy on the average in the results to lengthy computations. The general trend in computation systems is to use normalized floating-point arithmetic and to use other methods of checking on the significance of the answers.

20.2.5. Decimal vs. Binary. Another method of distinguishing arithmetic systems is to check as to whether they operate in the binary number system or in the decimal number system. In most decimal number systems there is a binary representation of the individual digits. This binary representation may be in the true binary number system or it may be in variations of this where other sequences of binary bits represent the corresponding decimal digits.

In four-channel serial by digit systems there has been a tendency to design decimal arithmetic systems. The actual arithmetic operations are more complicated, first because the actual addition of individual digits requires more equipment, and processes such as multiplication and division must be done by repeated additions or subtractions in corresponding positions, whereas in the binary system the additions and subtractions occur once in each position of the number. Thus arithmetic systems which are binary are simpler and tend to be faster than the corresponding decimal systems.

20.2.6. Command Sequencing. There are two methods of determining the sequence of commands that are to be executed in the computational process. In one case a serial counter counts out consecutive memory addresses so that the system operates by looking at consecutive memory positions and executing the commands found there. The other system involves placing with each command information which specifies the location of the next following command. In computing machines with so-called random access, such as in magnetic core memory systems, there is a tendency to use the serial technique since this makes more efficient use of memory space. On the other hand, in cyclic memory systems where information becomes available only once every cycle of the memory system there have been designs in which each command specifies the location of the next following command. This makes it possible to execute commands at a higher rate, since each command can appear in a location in memory such that it is available for execution as soon as possible after the completion of the preceding command.

20.3. SPECIAL CHARACTERISTICS OF MAGNETIC DRUM COMPUTERS*

Because the cost of storing information on magnetic drums is much less than for storing in other memories suitable for use on high-speed computers, it is possible to construct certain registers as part of the magnetic drum memory.

20.3.1. Buffer Memories. Placing two heads on the same track some number of sector lengths apart provides a lower-capacity memory than that of a full channel but with a proportionally lower access time. If there are normally 100 sectors around the drum, then two heads spaced 10 sector lengths apart may be used to provide a 10-word capacity loop with an access time of one-tenth the access time to a track containing but one head.

A section of a drum rotating past two heads is depicted in Fig. 20.1. Information is read from one head, amplified, and stored for one bit time in a flip-flop. It is then recorded by a head set 10 sector spaces back along the circumference (actually 10 sector spaces less one bit space).

As before, if the word is stored serial-serial, the sector length will be the same as the word length. If the storage is series-parallel, then as many of these loops as there are bits in parallel are needed to store the 10 words.

The access time to the loop is reduced since it requires only a fraction of a drum

* Sections 20.3 through 20.4.2 by William R. Arsenault.

revolution for a particular bit of a word to appear in the output successive times. If faster access is desired the heads may be spaced closer together. Some computers using this method of storage have several loops of different head spacing.

Since the information recorded is of interest only while it exists between the two heads, several other independent loops may be spaced around the circumference on the same track. The record head associated with the loop erases all information previously recorded on the track and inserts the new information. Therefore, information recorded by heads associated with other loops will not interfere with each other.

The loop may be addressed in a manner similar to any read-record head. The head is selected and a coincidence with a sector address is obtained.

20.3.2. Circulating Registers. Information contained in a register is not used until it appears at the end of the register, i.e., as it is shifted out. In a serial-binary machine

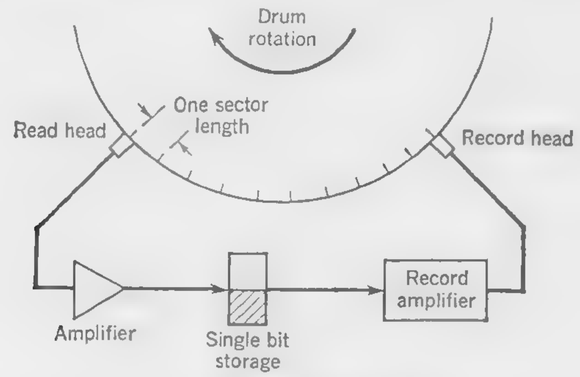


Fig. 20.1. Recirculating register.

addition may be performed by adding the least significant bits of the two operands, then shifting and adding the next two significant bits. The process continues until the two words are added together.

This operation can be made to take place at the machine clock rate and in synchronism with the machine word times. The same results may be obtained if the registers take the form of circulating loops on the drum. Instead of the head spacings being multiple word lengths apart, they now must be something less than one word length apart.

For such loops the two heads must be spaced very close together. For a machine with a word length of 40 bits and a bit density of 100 per inch, the heads must be spaced less than 0.40 in. apart. This puts a physical limitation on the size of the head; however, several are commercially available that can meet these specifications.

Putting the storage registers on the drum complicates the logical design of a computer but realizes great savings in equipment. For a serial-binary machine up to 80 per cent of the equipment associated with the shifting registers may be eliminated as compared with equivalent vacuum-tube or transistor registers.

20.3.3. Loop Realization and Machine Timing. Successful operation of circulating registers takes advantage of the complete synchronization that exists in a magnetic drum processor. Figure 20.2 depicts a section of a drum and schematically shows some logical elements associated with two tracks. The first, the accumulator, is a circulating-loop register, where two bits of the word contained circulate through static flip-flop storage. Normal information flow is from A_{101} to the record amplifier and back onto the drum. The second is an information track with a single head associated with it. The particular head is selected by addressing in a manner described in earlier sections.

The bits of the two words in question are shown in position at the beginning of a word time. Assume the command being executed calls for an addition of the word in the accumulator and the word N of the selected track. The record amplifier will be

made to select and record the output of the binary adder. This output is the result of adding the bits contained in A_{101} and A_{200}, the least significant bits of the respective words.

The b_0 at the record head is then the first bit of the new word being generated by the add command. As the drum rotates under the head successive bits will be added and the results recorded in the accumulator. At the end of the word time the record amplifier again selects the output of A_{101} so that the accumulator word will recirculate.

If it is desired to insert a new word into the accumulator from the memory, the record amplifier will select the output of A_{200} directly for the correct word time.

Again, at the beginning of a word time, b_0 of the word on the information track appears in A_{200}. This implies that this information, as it appeared on the drum,

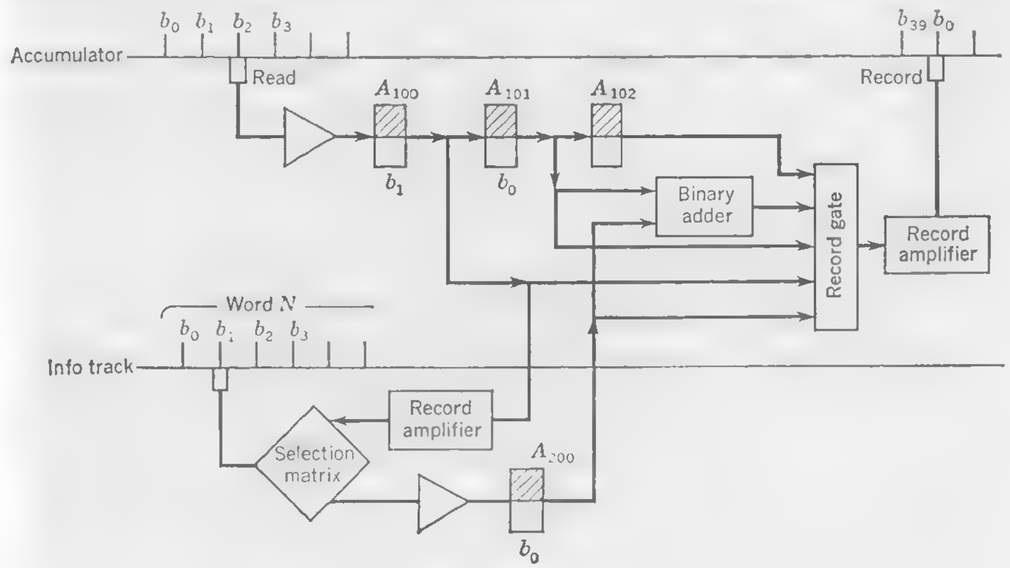


Fig. 20.2. Arithmetic unit.

must have already passed under the head, been amplified, strobed, and set into A_{200}. If it had been desired to record new information into word location N, then b_0 must have already been recorded. It follows then that there is a one bit delay between reading and recording with the same head.

In order to send information from the accumulator to the memory, it is necessary to start the record operation one bit time earlier. This is the purpose of A_{100}. When b_0 appears in A_{100} the read-record head is in the correct position to record it. This type of compensation is not necessary when transferring information between several loops that may exist in a machine.

The binary adder shown in Fig. 20.2 is only part of the logical circuits that may exist. Such operations as compare, extract, etc., may all take place in a word time.

20.3.4. Shift Operations. The relative position of bits must be defined at a particular time when using these dynamic techniques. For purposes of discussion, this will be defined as the beginning of a word time. Thus b_0 is in A_{101} at the beginning of a word time.

In order to shift information within a word it is necessary to shorten or lengthen the loop. If it is desired to shift left, the loop is lengthened 1 bit to 41 bits. This is accomplished by inserting A_{102} into the information flow. The bit b_0 has 41 positions to assume but only 40 bit times to assume them. Therefore, at the beginning of the next word time, b_0 will be in A_{100} or one bit space to the left. For each word time that the information is made to pass through A_{102}, the respective bits will be one more position to the left.

In a like manner, information may be shifted right by shortening the loop. This is accomplished by passing A_{101}. Now the loop is only 39 bit positions long and b_0 will

already be recorded back on the drum after one word time. The bit b_0 will occupy the space formally occupied by b_{39}.

The entire computer operation may be designed around this technique. The exception is when it is necessary to input or output to or from an asynchronous device, such as a tape punch. Here, some amount of buffer storage must be provided.

20.4. SOME GENERAL-PURPOSE DRUM COMPUTERS AND THEIR CHARACTERISTICS

20.4.1. The IBM 650. *Memory.* This machine uses a cobalt-nickel-plated drum, 4 in. in diameter and 14 in. in length. The drum rotates at 12,500 rpm. The recording density is 50 bits per inch, giving a machine pulse rate of 128,000 pps.

Digit					
0	0	1	1	0	0
1	1	1	0	0	0
2	1	0	1	0	0
3	1	0	0	1	0
4	0	1	0	1	0
5	0	0	1	1	0
6	1	0	0	0	1
7	0	1	0	0	1
8	0	0	1	0	1
9	0	0	0	1	1

FIG. 20.3. Five-bit drum code.

Information is stored in a series-parallel form, 5 bits in parallel, representing a decimal digit. A word is 10 decimal digits long, plus a digit for sign. An additional space bit separates successive words. A word, then, occupies 5 tracks (referred to as a band) by 12 bits in the circumferential direction. Each band contains 50 words, and 40 bands give a capacity of 2,000 words.

The drum stores information in a 5-bit code shown in Fig. 20.3. This code can easily be checked for error since each decimal digit always contains two and only two binary 1's. The digits are converted between this code and a biquinary code when the word is transferred between the drum and the logical circuitry.

Word Structure. The machine word is 10 decimal digits long, plus sign. Information, when in the logic section, is carried in the word in the form of a 7-bit biquinary code.

When a word in the machine takes the form of a command the digit positions have the significance shown in Fig. 20.4. This is the so-called modified single-address code.

The two most significant digits define the operation that is to take place. There are 44 possible operations in the machine. The operand address part of the command word defines the address on the drum where data are to be found or sent. When an

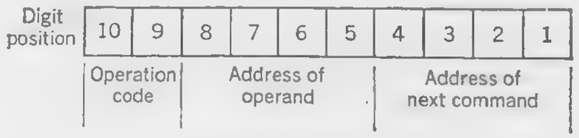


FIG. 20.4. Command word.

operation does not imply an operand address, this part of the word may take on one of the following meanings:

1. Number of positions to be shifted either right or left
2. Address of a block of words in the memory to be output
3. Address of a block of words where data are to be input
4. Address of an alternative command, dependent upon certain tests
5. Address of a word where a search operation is to start

The least significant four digits specify the address of the next command unless a branch is indicated (item 4 above).

Logic. The logical section of the computer consists of two registers, the distributor and the accumulator. In addition, there is the program register that holds the command.

These registers are based on capacitor storage and circulate continuously in synchronism with the drum, much as circulating registers discussed in Sec. 20.3.

The distributor acts as an intermediate storage link between the accumulator and

memory. Any number transferred between the accumulator and the memory passes through the distributor.

The accumulator has a capacity of 20 digits. It is divided into two 10-digit halves; however, carries propagate between the lower and upper halves. A 10-digit operand may be added to the 20-digit accumulator to form a 20-digit sum. A 10-digit operand may be multiplied to form a 20-digit product, or divided to form a 10-digit quotient and 10-digit remainder.

The distributor and both halves of the accumulator are addressable as well as memory locations on the drum. This allows either half of the accumulator to be added to or subtracted from the other half; either half added to itself; the upper half to be multiplied by itself, i.e., squared; etc.

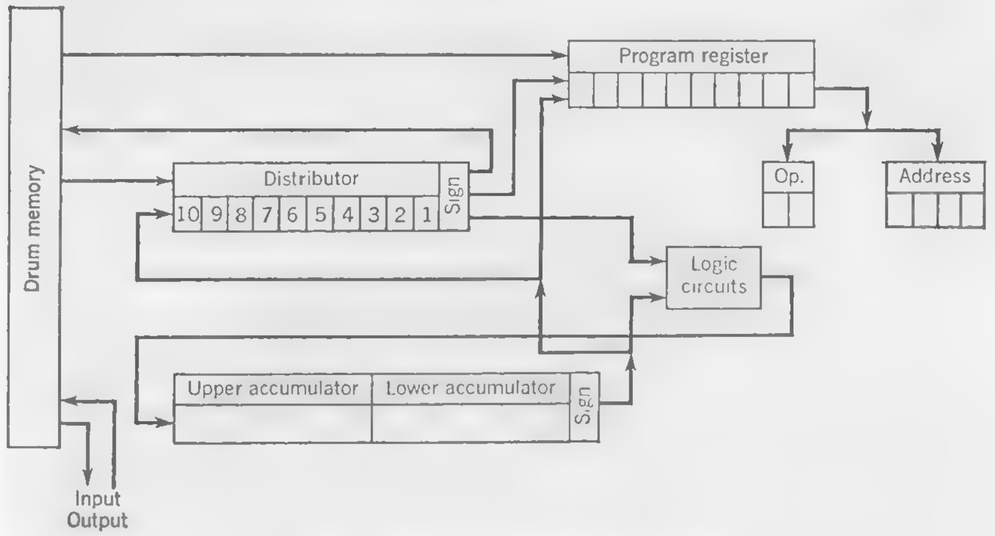


FIG. 20.5. Information-flow block diagram.

Among the many operations common to a general-purpose computer, the machine is capable of performing a table-look-up function. The argument is stored in the distributor and the location where the search is to start is given in the operand address. The machine then compares successive words on the drum with the word in the distributor. When the word in the memory becomes equal to or larger than the word in the distributor, the operation terminates and the address of the compared word is given in the accumulator. There are 50 words around the drum. If no comparison is made by sector 48, the machine switches to the next band and searches that band, starting at the first sector. Sectors 49 and 50 are ignored in the search, and this time is used to switch bands.

Command Execution. Information flow within the computer takes place in a series-parallel fashion. All logical operations take place by passing the two operands serial by digit through the logic circuits. The word in the accumulator is merged with the operand word to form the desired results. The result is put back into the accumulator.

The registers of the machine circulate in a dynamic fashion, in synchronism with the drum timing. Since the accumulator is two word lengths long it takes two word times for information to circulate. Information in the lower accumulator is synchronized to enter the logic circuits during an even word time and the upper accumulator during an odd word time.

Shifting is performed by shortening or lengthening the number of digit positions in the accumulator. To shift left, the accumulator is lengthened one digit position. Each circulation of the accumulator moves the digits one position. Thus, it takes two word times per bit position shifted to execute this part of the operation.

A typical add command takes 8 word times to execute. Assume that an add to lower accumulator command (AL) is located in memory address 0001. During this

word time the command is gated from the drum into the program register. During word time 0002, the operation part of the command is sent to the operation register and the operand address is sent to address register. During word time 0003, the appropriate drum heads are selected. The machine then waits for the operand sector to pass under the heads.

If the operand is in one of the 40 word locations of sector 04, there will be no wait and the data word is sent to the distributor. It was stated that the lower part of the accumulator circulated in a manner such that it could enter the logic circuits only during an even word time. The machine must then wait until sector 06 to add the two numbers. The upper part of the accumulator passes by during sector 07 time. Meanwhile, the address of the next command is sent to the address register. It is therefore possible to locate the next command in one of the words at sector 09.

With the data at the optimum location the entire command requires 0.768 msec.

Multiply is carried out by repeated additions and shifts and requires from 20 to 200 word times to perform the arithmetic functions. Divide requires from 60 to 240 word times to perform the arithmetic.

Assuming random data and instructions the following times are quoted for command execution:

Command	Average Time, msec
Add, subtract, store	5.2
Shift	2.8
Branch	2.7
Multiply	11.6
Divide	14.5

Equipment. The processor is composed of some 2,000 tubes and 3,600 crystal diodes. Tube types used are the 5965, 6211, 12AY7, 6AL5, 2021, and 5687. The machine consumes 16 to 18 kva of power.

Auxiliary Equipment. Normal input and output are via punch card. A high-speed core storage of 60 words may be added. This storage is addressable from the processor and also acts as a buffer between the machine and magnetic-tape units. Up to 60 magnetic-tape units may be tied to the machine.

20.4.2. The Royal-McBee LGP-30. *Memory.* The drum memory is approximately 8 in. in diameter and 10 in. long. Words are stored in a pure serial form, 32 bits to the sector. The 64 sectors per track give a total track capacity of 2,048 bits.

The drum rotates at approximately 3,500 rpm, giving a bit rate of 120,000 per second.

There are 64 tracks to the main memory. This provides a total storage capacity of 4,096 words.

In addition to the storage above, there are three circulating registers. These are called the counter register, instruction register, and the accumulator. The registers operate as discussed in Sec. 20.3. Here it is required that the heads be spaced 31 bits. At a density of 80 per inch, the head centers are approximately 0.4 in. apart.

Word Structure. The machine word is 30 bits long, plus a single bit for sign. A thirty-second bit is used as a space bit on the drum.

Information is carried in pure binary form with the decimal point considered ahead of the most significant bit. Numbers vary from $+1$ to -1 in increments of 2^{-30}.

When the machine word is used as a command only 16 of the bits have significance. Four are used to define the operation and the remaining 12 define the address of the operand. This is referred to as a single-address command structure.

Order Code. The complete order code is included here as it represents a near minimum number of operations needed to perform problems conveniently on a general-purpose computer. The code is the four binary bits carried in the command word that define what operation is to be performed. An operand address is needed for each command word except for the last three commands in the list.

Logic. There are three registers in the machine used to carry out all logical operations. These three registers are circulating types where a majority of the information held in the registers is contained on the drum.

One circulating register is called the *counter*. It is one word length long and stores

the address of the next command. As previously described, single-address machines usually obtain their commands from successive addresses. Each time this address is used it is augmented by one and held until it is used to designate the next command. This procedure continues until a transfer command inserts a new command address in the counter.

A second circulating register is called the *instruction register*. This normally holds the command word containing the instruction and operand address. For this purpose information is derived directly from the main drum storage.

Table 20.2. The LGP-30 Command List

INSTRUCTION ORDER LIST SHOWING CODE FOR EACH INSTRUCTION

Code	Effect
Arithmetic:	
0001	Bring. Clear the accumulator, and add the contents of location m to it
1110	Add contents of m to the contents of the accumulator, and retain the result in the accumulator
1111	Subtract the contents of m from the contents of the accumulator, and retain the result in the accumulator
0111	Multiply the number in the accumulator by the number in memory location m, terminating the result at 30 binary places
0110	Multiply the number in the accumulator by the number in m, retaining the least significant half of the product
0101	Divide the number in the accumulator by the number in memory location m, retaining the rounded quotient in the accumulator
1001	Extract, or logical-product order—i.e., clear the contents of the accumulator to 0 in those bit positions occupied by 0's in m
Transfer control:	
1010	Transfer control to m unconditionally—i.e., get the next instruction from m
1011	Test, or conditional transfer. Transfer control to m only if the number in the accumulator is negative
Record:	
1100	Hold. Store contents of the accumulator in m, retaining the number in the accumulator
1101	Clear. Store contents of the accumulator in m and clear the accumulator
0010	Store only the address part of the word in the accumulator in memory location m, leaving the rest of the word undisturbed in memory
0011	Return address. Add 1 to the address held in the counter register C and record in the address portion of the instruction in memory location m. The counter register normally holds the address of the next instruction to be executed
Miscellaneous:	
0100	Input. Fill the accumulator from the Flexowriter
1000	Print a Flexowriter symbol. The symbol is denoted by the track number part of the address (x)
0000	Stop. Contingent on five switch (T_1 . . . T_5) settings on the control panel

The third register is the *accumulator*. This register is one word length long for most operations. However, a second read head spaced an additional word length, plus a bit space, allows the register to be lengthened to 65 bits during certain portions of the multiply and divide operations.

The execution of a command is divided into eight steps or phases. For most operations only four are used.

In the first phase a search is made for the command word whose address is held in the counter register. That part of the address designating the track is sent to the track-selection register, comprised of six toggles, and a sector search is made on the remaining bits.

Upon sector coincidence the machine is set to phase 2 and the command word is read into the instruction register. This period always lasts a single word time.

Again during phase 3 a search is made, this time for the operand address. Upon sector coincidence the machine is set to phase 4.

Phase 4 lasts one word time and is the period when all logical and arithmetic operations take place, except for the multiply and divide orders. If the operation is addition, the word in the accumulator is added serially, bit by bit, to the word being read from the memory. If it is a send order, the word in the accumulator is recorded in the sector found during phase 3.

For a transfer order, the contents of the instruction register are sent to the counter register during phase 4. This, then, becomes the address of the next command, as required.

Phases 2 and 4 are each one word time long and phases 1 and 3, the search periods, are a minimum of one word time. Assuming the operand is located in the optimum position, these operations can be completed in four word times.

Multiplication. Multiplication is carried out by a series of additions and shifts. During phase 4, the multiplicand, which is specified by the operand address, is read from the memory into the instruction register. The multiplier, which already exists in the accumulator, continues to circulate. During the next two periods the product

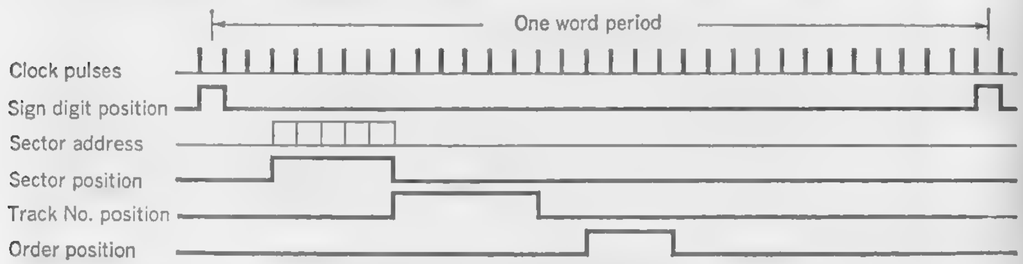


Fig. 20.6. Drum timing signals.

is formed. First, the accumulator is extended to 65 bits by reading from the second head rather than from the first. This lengthens the loop to slightly over 2 words in length. The effect of this is to precess the information held in the accumulator one bit position each two word times; i.e., the bits of the word are all shifted one position in the more significant direction each complete circulation.

The accumulator then holds the multiplier and the partial product. As the accumulator circulates, the multiplicand is added in or not, depending upon the value of the pertinent digits of the multiplier. This addition operation happens each full accumulator circulation. Also, the accumulator "shifts" one bit each circulation. The multiplicand is therefore added into a successively lesser significant part of the partial product.

The partial product is initially one word length long and progressively grows. As each digit of the multiplier is used it is dropped from recirculation. This compensating storage requirement allows the accumulator to handle both pieces of information while it is extended to two plus word lengths.

This operation requires some 64 word times, plus the initial four minimum word times required for phases 1 through 4, or approximately 18 msec.

Machine Timing. There are several timing tracks permanently recorded on the drum. These are used to generate various timing and gating signals that occur within a word time.

The first line of Fig. 20.6 shows the clock or bit pulses. These pulses occur at a rate of 120,000 per second. The second line shows a signal occurring at the sign digit position. Whenever it is desired to define the existence of the sign of a word, say in the accumulator, the coincidence of this signal and the output of the accumulator loop will produce the needed information.

A third track on the drum defines the various sectors around the drum. On this track are permanently recorded sequential numbers of the 64 sectors around the drum. As actually utilized, the numbers on this track represent the *next sector* that will pass under the head. A fourth track is used to define the bits where the sector part of the address occurs.

A fifth track defines where the channel part of the address occurs, and a sixth defines where the order bits occur.

Address Selection. It was discussed above how, in phase 2, the command word was read from the memory into the instruction register. Previous to this it was necessary to select the proper track and sector defining the location of this word.

Given in the counter register is the address of this new command. This word is circulating in synchronism with all the information on the drum and in particular with the timing tracks. First the timing signal from track 4 is gated with the output of the counter register. This selects out the correct 6 bits to send to the track-selection register, which in turn selects the correct head.

At the same time the timing signal from track 4 is used to gate information from track 3 and the counter register. The contents of these bits from the counter register are compared with the output of track 3. If they compare exactly, then the next sector, the start of which is defined by track 2, is the desired one and thus contains the new command word. This word is then gated to the instruction counter.

If a coincidence is not found a comparison is made during the correct bits of the next sector. This continues until the correct sector is found.

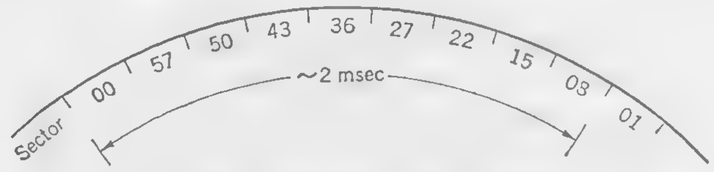


Fig. 20.7. Sector interlacing LGP-30.

Sector Interlacing. The sector addressing around the drum is such that there are 9 sectors between successive addresses. This is shown in Fig. 20.7. Assuming a command in sector 00, the operand search begins during sector 57 time. The operand may be located in any of the words associated with sectors 50, 43, 36, 27, 22, or 15. The search for the next command begins at sector 08 so that the next command that is located in 01 will be read without delaying a drum revolution.

This procedure can take place for all those commands which require only the first four program periods. Thus execution time including access can be as low as 2 msec.

Equipment. The computer uses a total of 110 vacuum tubes and 1,400 diodes. The power requirement is 1.5 kw at 115 volts alternating current.

Auxiliary Equipment. A Flexowriter is used as input-output. This is a combination typer and paper-tape reader-perforator. Since the registers are dynamic they are not readily displayable. An oscilloscope is provided to view the contents of the three registers.

20.4.3. The Bendix G-15 Computer. *Memory.* The memory of the Bendix G-15 computer is a magnetic drum. The main memory consists of 20 tracks with 108 words in each. There is a 16-word high-speed memory in the form of four tracks with four words each. The arithmetic circuits involve four tracks, three with two words and one single-word register. Another single-word register is used in the control circuits and there are timing tracks on the drum.

As indicated in Figs. 20.9 and 20.11 the G-15 has a source-destination structure (see Sec. 20.2) and the command essentially specifies the source and destination involved and the time at which transfer is to occur. In order to provide for minimum access for commands, each command also specifies the time at which the next command is to be read from memory.

All tracks in the memory recirculate, making it possible to transfer information from any track to any other track. In terms of Fig. 20.9 there are direct connections (not shown) from each read head to the corresponding write head, 2.

Inverting Gates. All transfers of information are via the inverting gates (8 of Fig. 20.9) and under control of parts of the command (*CH* and *S/D* of 7 of Fig. 20.11). During such transfers the sign bit may be deleted (absolute value), the number may be complemented if negative, or the sign may be changed and then complemented if

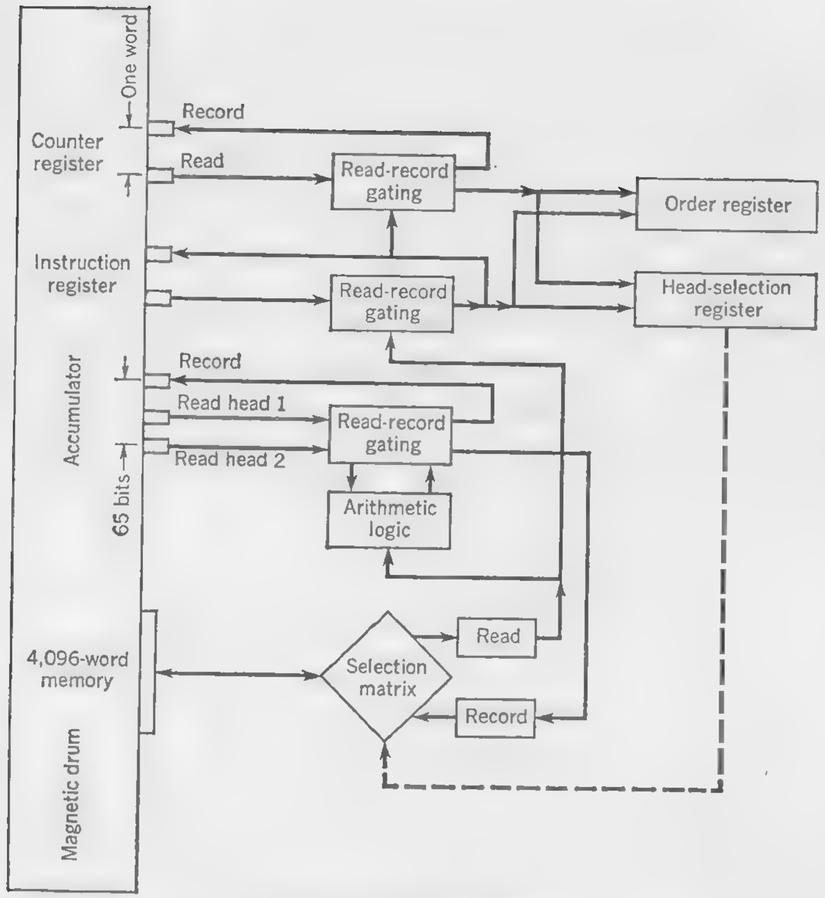


Fig. 20.8. LGP-30 block diagram.

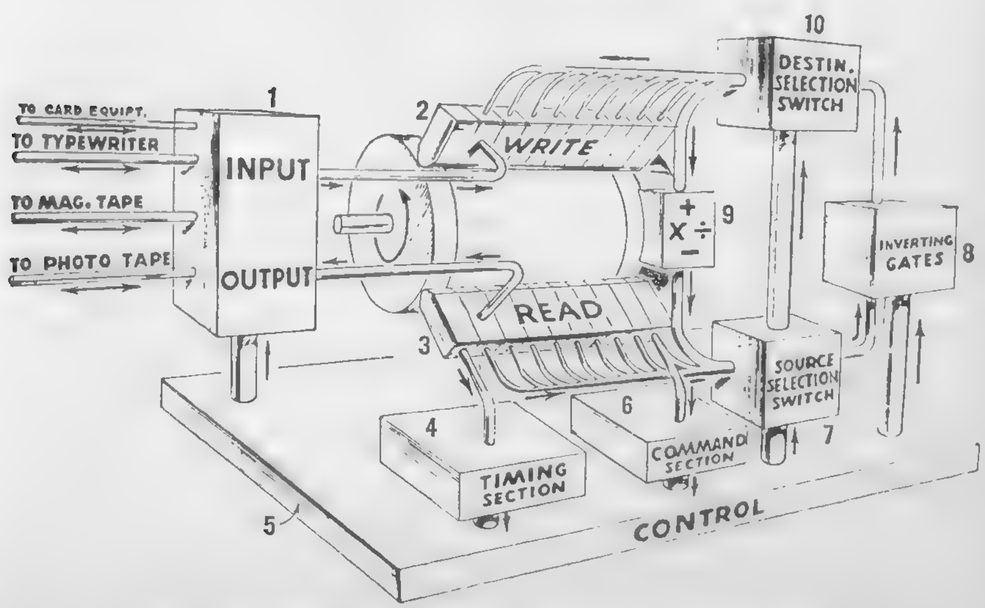


Fig. 20.9. Bendix G-15 diagram. (*Courtesy of Bendix Corp.*)

negative. All these operations may be done on the basis of single precision (29-bit numbers) or double precision (58-bit numbers).

Arithmetic Operations. The arithmetic circuits include three two-word registers and a single one-word register. One of the two-word registers and the single-word register have addition circuits associated with them. Thus there are two independent accumulators, one for single-length words and one for double-length words. Multiplication and division as well as shifting are done in the two-word registers. Under control of the command any number of digits in the multiplier may be used or any number of digits in a quotient may be developed. The particular arithmetic operation to be done depends only upon source and destination values of the command being executed.

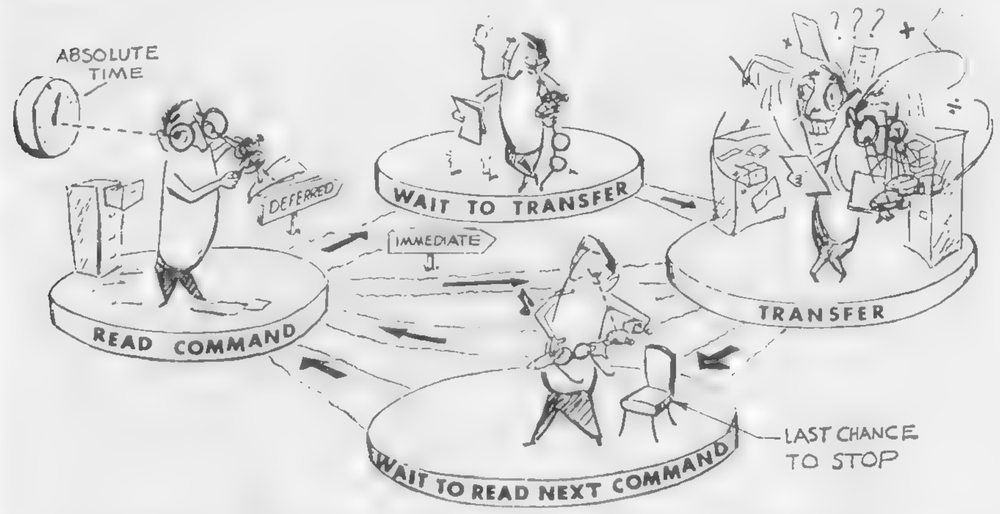


Fig. 20.10. Bendix G-15 command cycle. (*Courtesy of Bendix Corp.*)

Input and Output. One of the 108-word memory lines serves as a buffer register in the input and output process. Inputs and outputs are initiated by the control circuits upon the execution of the appropriate command. Input terminates upon reception of a STOP code from the input device. Output terminates after a few characters under format control or after larger numbers of words when the buffer empties. Thus the maximum number of words which can be handled as a block on input or on output is 108 words.

Input is from the typewriter and a photoreader which reads punched paper tape. Output is to the typewriter and to punched paper tape. Auxiliary equipment provides communication to and from magnetic tape and punched cards.

Once an input or output process has been initiated the computer may continue with the program. The program may test for the termination of an input or output process in order to know when input information may be used.

The Command Cycle. The command circuits usually pass through four states illustrated in Fig. 20.10. The states are (1) READ COMMAND, (2) WAIT TO TRANSFER, (3) TRANSFER, and (4) WAIT TO READ NEXT COMMAND. The WAIT periods are caused by the cyclic nature of the memory. Note that in the case of "immediate" commands the WAIT TO TRANSFER period may be by-passed. Immediate commands may transfer for any number of word times up to a maximum of 108.

Precession. Figure 20.11 shows that information from the source passes through the "early bus" to the inverting gates and through the "intermediate bus" to a box where there is a possibility of the information passing through the one-word *AR* register before arriving on the "late bus." Under control of one command as many as 108 words may be transferred in this way and the effect is to displace all the words one word position. In other words, their sector location has been increased by unity or their address has been increased by one.

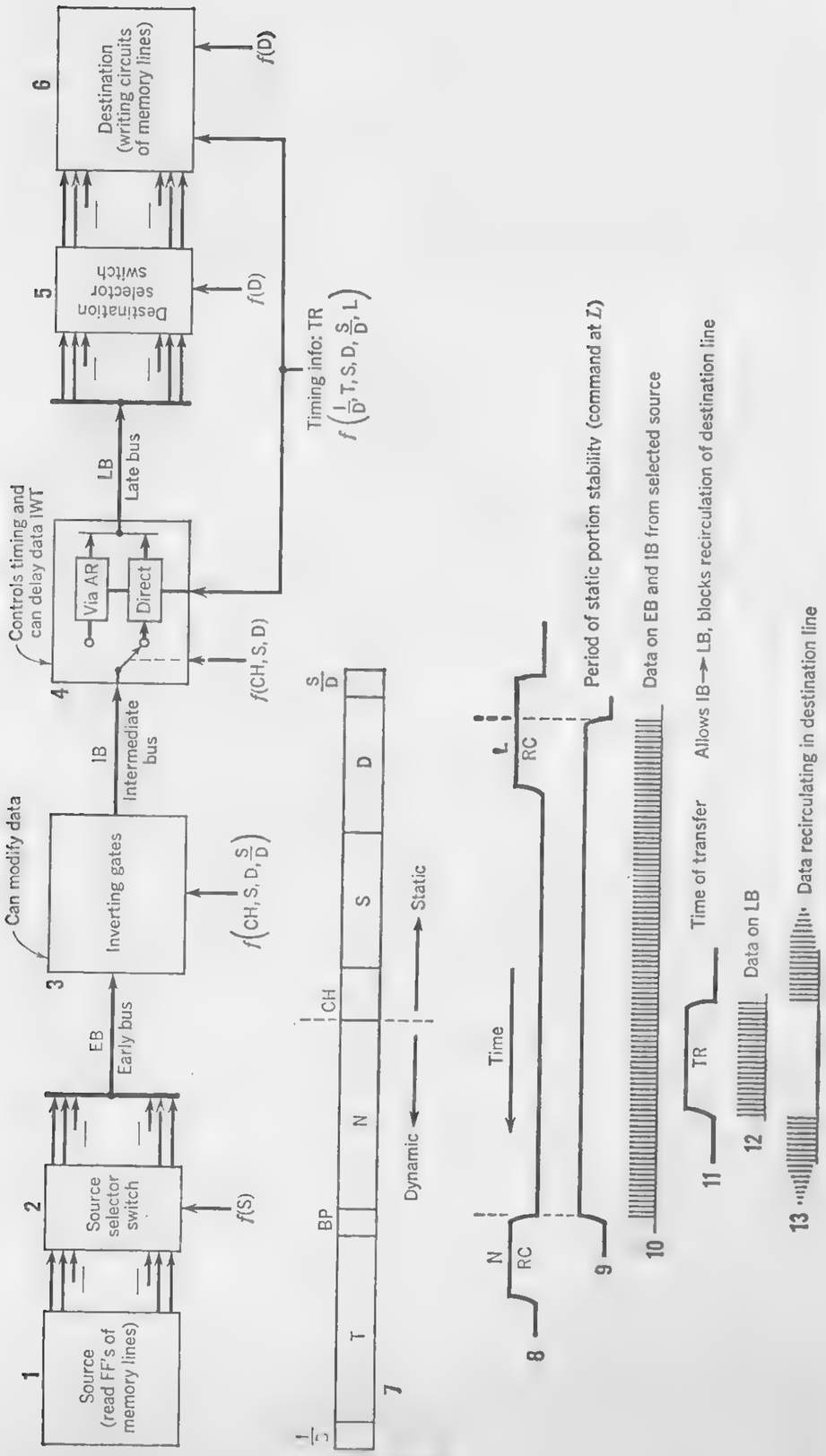


FIG. 20.11. Bendix G-15 transfer system.

There are precession circuits associated with the input and output buffer which give precession by four bits or by four words. These operations can be programmed to occur anytime and not just during an input or output process.

The Digital Differential Analyzer Attachment. An attachment to the G-15 may be had which converts part of the computer into a digital differential analyzer (see Sec. 19). This conversion uses part of the memory and does not interfere with the operation of the general-purpose computer. This makes it possible to control the integrations going on in the *DDA* "on the fly," so to speak. For example, it is possible to change the system of equations being integrated without interrupting the integration process.

20.5. LARGE-SCALE DIGITAL COMPUTERS

In the digital-computer field a certain stability has been reached in magnetic-drum-size computers in that a number of designs have been developed and marketed and have failed commercially, whereas a few designs are being relatively widely sold and show promise of surviving for a time. On the other hand, in the large-scale high-speed class there are a number of new entrants in the field and their products have not had a real test in the computer market. Therefore, instead of describing particular computers as was done in the case of magnetic-drum computers a composite device will be described.

20.5.1. Memory Hierarchy. *High-speed Memory.* The high-speed memory is of the coincident-current type using magnetic cores to store the information. The memory-cycle time is 2 to 10 μsec and the memory size is from about 4,000 to approximately 32,000 words (numbers). Usually the memory comes in blocks of about 4,000 and the customer may have a memory of any intermediate size that he desires. The memory is parallel in nature with all the binary bits which represent a "word" being available simultaneously.

Some computing systems have a "read-only" memory (also called "dead memory" or "reserve memory") which has the same performance characteristics as the high-speed memory but cannot be written into. Such "read-only" memory is used to store subroutines and programs which are frequently used. For example, trigonometric functions or an assembly program designed to facilitate the insertion of programs may be in this memory.

Medium-speed Memory. The medium speed consists of one or more magnetic drums each holding perhaps 20,000 words. Random access to items here takes from 5 to 10 msec but character-transfer rates may be 100 kc (in contrast to megacycle rates in the high-speed memory).

Slow-speed Memory. The slow-speed memory consists of a number of magnetic-tape units each holding a 2,400-ft reel of tape. The tape is wide enough to record one character in parallel including any control or checking pulses required by the system. Random access to an item is on the order of minutes, but character rates may be above 100 kc. The computing system has sufficient control and buffering equipment so that magnetic-tape operations are only initiated by the computer and then computation may proceed while transfers are being completed.

General Memory Organization. In most existing systems the organization of the levels of memory is determined by the program executed in the computer. Thus the organization in this case depends either upon the programmer or upon a compiler which automatically allots storage space and generates the required program.

In some systems blocks of data are moved from one level of storage to another depending upon their frequency of use or upon length of intervals of disuse. Alternatively, in a random way blocks may be moved from high-speed levels to lower-speed levels to make room for new blocks. Systems organized in this way are adaptive in nature since in large-scale data processing problems, items with periodic (seasonal) fluctuations in "duty factor" will generally be located in that level of memory which leads to the optimum processing conditions. As changes in activity occur, the memory is reorganized accordingly.

20.5.2. Arithmetic System. The basic arithmetic operations are available in single-precision and double-precision binary floating-point as well as in fixed-point integral

form. The fixed-point operations facilitate subscript-type operations and are useful in data-processing problems. On the average, arithmetic operations can be done at the rate of 100,000 per second.

20.5.3. Control System. There is a conventional set of arithmetic commands as well as transfer of control and conditional transfer of controls which depend upon the signs of certain numbers or upon whether or not they are zero.

Interrupt facilities make it possible for communication with terminal equipment to proceed with a minimum of delay to the main program. For example, a typewriter may have been requested to type a block of data from the high-speed memory. Since the unit can accept characters only once every 100 msec, it gives out a "ready" signal when it desires a character; the computer is "interrupted" and supplies the desired character, and then the computation proceeds until the typewriter again needs a character.

It is possible to store "markers" with the data and use this same interrupt facility to terminate iterative loops in the program in a way which minimizes the number of commands in the loop and consequently gives maximum speed.

20.6. A SIMPLE ONE-ADDRESS COMPUTER

The behavior of a digital electronic device may be indicated by producing a block diagram showing the possible paths of information transfer and by preparing a

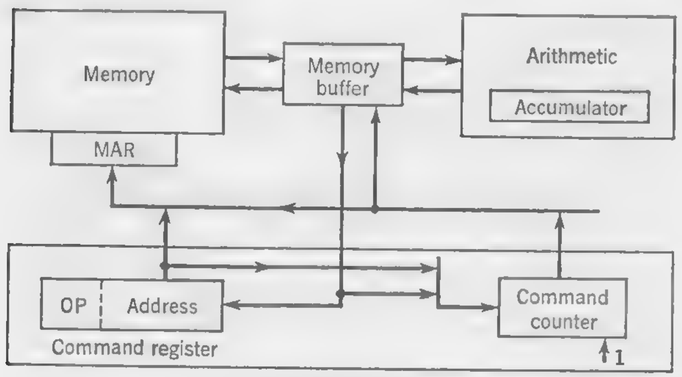


FIG. 20.12. A one-address computer.

sequence scheme which indicates the timetable which such events may follow. The block diagram serves as a roadmap indicating the possible paths of transfer and the sequence diagram indicates under all circumstances the actual timing of events which take place. The various entries in the sequence diagram are called "statements"; and these may be indicative statements indicating that certain action takes place, declarative statements specifying the state of events under certain circumstances, or interrogative (or conditional) statements which indicate two possible sequences of events depending upon the status of some particular indicator in the system. In general, the structure of this language is very similar to the problem-oriented language called ALGOL.

Generally, indicative statements have a form in which one item replaces another item. Parentheses around quantities are used to indicate the content of that quantity. The quantity inside the parentheses plays the role of a name. A conditional statement always consists of two parts; the first clause asks the question and the second part states the alternatives. The question terminates with the question symbol, and the statement after the question mark terminates with a standard punctuation symbol such as a comma, semicolon, or period. If condition statements are nested, braces may be used to enclose the inner statements in a form similar to normal English usage.

Certain points in a sequence diagram may be labeled as entries by writing the name of the entry followed by a colon.

A simple one-address computer will be described in terms of a block diagram (Fig. 20.12) and a sequence diagram (Fig. 20.13). The system consists of a memory, which appears in the upper left-hand corner of Fig. 20.12; an arithmetic unit, which appears in the upper right-hand corner; and a memory buffer register (upper center), which serves as a communication path between the various units and the memory. On the bottom of Fig. 20.12 is the control unit with a command register on the left-hand side and a command counter (CC) on the right-hand side. Information is

0: (CC) → (MAR),
1: (CC) + 1 → (CC),
2: ((MAR)) → (MB) → (CR),
3: A → (MAR),
4: Is OP operand type? ((MAR)) → (MB);
5: OP?

5.1.	CA (Clear Add):	(MB) → (AC), 0.
5.2.	AD (Add):	(AC) + (MB) → (AC), 0.
5.3.	CS (Clear Subtract):	−(MB) → (AC), 0.
5.4.	SU (Subtract):	(AC) − (MB) → (AC), 0.
5.5.	MU (Multiply):	(AC) × (MB) → (AC), 0.
5.6.	DI (Divide):	(AC) / (MB) → (AC), 0.
5.7.	ID (Inverse Divide):	(MB) / (AC) → (AC), 0.
5.8.	ST (Store):	(AC) → (MB) → ((MAR)), 0.
5.9.	TC (Transfer Control):	A → (CC), 0.
5.10.	TZ (Transfer Zero):	(AC) = 0? [TC.], 0.
5.11.	TP (Transfer Positive):	(AC) > 0? [TC.], 0.
5.12.	TN (Transfer Negative):	(AC) < 0? [TC.], 0.
5.13.	TNZ (Transfer Not Zero):	(AC) = 0? [0.],[TC.].
5.14.	TNP (Transfer Not Positive):	(AC) > 0? [0.],[TC.].
5.15.	TNN (Transfer Nonnegative):	(AC) < 0? [0.],[TC.].
5.16.	SUB (Subroutine Transfer):	O → (MAR), (CC) → ((MAR)), A → (CC), 0.
5.17.	IT (Indirect Transfer):	(MB) → (CC), 0.

A: Address
AC: Accumulator
CC: Command Counter
MAR: Memory Access Register
MB: Memory Buffer
CR: Command Register
OP: Operation

FIG. 20.13. Sequence scheme for simple one-address computer.

supplied to both the command register and the command counter from the memory buffer; it is also supplied to the command counter from the address portion of the command register.

The memory may store as many as, say, 10,000 numbers in locations which are numbered 0, 1, 2, and so on up to 9,999. In order to specify which particular location is involved in a transfer to or from the memory buffer (MB) there is a memory access register (MAR). This register must store some number between 0 and 9,999, and this determines which particular number in the memory, or instruction in the memory, will be transferred to the memory buffer.

In the arithmetic unit of Fig. 20.12, an accumulator is shown. In the following discussion it is not particularly important whether the arithmetic is in the binary or decimal number system, or whether it is in fixed-point or floating-point (and if fixed-point whether it is in integral form or in fractional form). In a practical computer the type of arithmetic may have some influence on the actual commands that are put into the computer; for example, in a floating-point system fixed-point arithmetic commands may be needed for the purpose of modifying commands.

The diagram of Fig. 20.12 is like a roadmap, in that it shows *where* information may go but it does not say anything about *when* the information is transferred. This timing information appears in the sequence scheme which is shown in Fig. 20.13. The general sequence of events is described in Fig. 20.13 as a cycle divided up into five parts and numbered 0, 1, 2, 3, 4, and 5. In the case of step 5 there is a large number of possibilities which depend upon the operation code of the particular command. For example, an addition operation appears in section 5.2 and a transfer of control is shown in 5.9. Note that step 4 is also optional in that some commands, for example, transfers of control, do not involve an operand; in the case of such commands nothing happens during the execution of step 4.

In order to read the information given in Fig. 20.13 it is necessary to explain the meaning of certain combinations of symbols. Generally, the elements in a sequence scheme are called "statements." So, for example, step 1 is the statement "increase the contents of the command counter by unity and put the result back into the command counter." (All the way through this discussion parentheses will be used to indicate "the contents of.")

Step 0 indicates that the contents of the command counter are transferred to become the contents of the memory access register. In step 2 the contents of the contents of the memory access register become the contents of the memory buffer. (This means that a number in the command counter was transferred into the memory access register; and the number stored in the memory, in the location which corresponds to that number in the memory access register, is the quantity which actually appears in the memory buffer.) After this the number is transferred from the memory buffer into the command register in the control circuits. In step 3 the address portion of the command, which is called A, is transferred to become the contents of the memory access register.

In step 4, if the operation code is such that an operand is required, the number in memory corresponding to the address now in the memory access register is transferred to the memory buffer. Note the punctuation in the specification of step 4. A question is asked, and if the answer to this is *yes*, the statement between the question mark and the first punctuation symbol (in this case a semicolon) is executed. If the answer to the question is *no*, this step is skipped and it continues with step 5.

In step 5 the operation code is inspected and, depending upon its value, there is a branch into one of 16 different sequences or statements as indicated in Fig. 20.13. The first seven possibilities in step 5 correspond to various arithmetic operations such as add, subtract, etc. In statement 5.8 there is provision for the transfer of information from the accumulator back into the memory. This is called a "store" operation.

As an example, consider statement 5.2, which is the addition operation, and note that it specifies that the contents of the accumulator are added to the contents of the memory buffer and the result is left in the accumulator. After this, there is a ",0." which is like a transfer-of-control instruction; in other words this means that the next event in the sequence scheme is to return to step 0. Returning to step 0 causes a new command to be obtained from the next following location in memory and the whole cycle repeats again. Note that the various arithmetic transfer statements corresponding to operation codes are each named by writing a name followed by a colon. For example, the addition sequence which was just discussed is labeled AD. If a colon follows a name it signifies that this name is a statement label.

In the transfer-of-control operation (5.9) the address in the command register is transferred to become the contents of the command counter, and the next event is step 0 of the sequence scheme. Statements 5.10 to 5.15 are conditional transfers; whether or not transfer occurs depends upon the number stored in the accumulator.

Next consider 5.13, which is a transfer on not zero. In the first part of the statement the number in the accumulator is inspected to see if it is equal to zero. If the answer to this is *yes* then the portion of the statement between the question mark and the comma (the periods inside the brackets do not count, since they are on a different level) is executed, which means that the next event in the sequence is the transfer to step 0. If the condition is not satisfied, in other words if the number in the accumulator is not zero, then the next event is statement 5.9, which is a transfer of control. The other conditional transfers operate in a similar manner.

Statement 5.16 is a subroutine transfer. This statement is somewhat more complicated than most of the other operation codes. The first thing that happens is that zero is placed in the memory access register; then the contents of the command counter are transferred into the memory buffer, and after that into the memory at location zero. All this is indicated by the statement (CC) → ((MAR)). The next step in the execution of this particular operation code is the transfer of the address A to the contents of the command counter and a return to step 1.

The indirect transfer (5.17) provides for a return from subroutines. This is an operand-type command; so by the time its execution takes place the number in memory corresponding to the address A has been transferred to the memory buffer. The next step in the execution is to transfer this quantity into the command counter and go back to step 1. Note that the memory buffer might be longer than the command counter; that is, there may be more digits in the memory buffer than in the command counter. In that case the statement of 5.17 would be interpreted as indicating that the corresponding digits of the memory buffer are to be transferred to the digits of the command counter. For example, if a subroutine has been entered by a subroutine transfer, then the current place in the main program has been stored in location 0 of the memory. An indirect transfer with address 0 would have the effect of transferring this number in location 0 of the memory back to the command counter via the memory buffer. This would restore the contents of the command counter to what was there prior to entering the subroutine. If a sequence of subroutines is to be nested then each successive subroutine must save the information in location 0 (by transferring it to some other memory location) in order to provide automatically the right transfers of control from the subroutines at one level to those of another level.

20.7. SEQUENCE SCHEME FOR ARITHMETIC OPERATION

Figure 20.14 shows the details of a typical list of arithmetic commands. In this presentation the supposition shall be that these commands operate in a floating-point decimal mode so that there is no problem about conversion of numbers from one number system to another, nor is there any difficulty about exceeding capacity in a normal computation.

A. Is OP arithmetic?, R.

A.1: CA (Clear Add):	(MB) → (AC), 0.
A.2: AD (Add):	(AC) + (MB) → (AC), 0.
A.3: CS (Clear Subtract):	−(MB) → (AC), 0.
A.4: SU (Subtract):	(AC) − (MB) → (AC), 0.
A.5: MU (Multiply):	(AC) × (MB) → (AC), 0.
A.6: DI (Divide):	(AC) / (MB) → (AC), 0.
A.7: ID (Inverse Divide):	(MB) / (AC) → (AC), 0.
A.8: ST (Store):	(AC) → (MB) → ((MAR)), 0.

AC: Accumulator
MAR: Memory Access Register
MB: Memory Buffer
OP: Operation
R: Return

FIG. 20.14. Sequence scheme for arithmetic operations.

In the case of all these arithmetic commands the next step in the process is the zero step in the sequence scheme (Fig. 20.13) as indicated by the "0." at the end of each of the statements A.1 through A.8.

Many computers do not have an inverse divide command; this means that in developing the value of an arithmetic expression it may be necessary to store temporarily

the contents of the accumulator so that it can be divided into some other partial result. The presence of the inverse divide command on occasion saves this extra store command and a location in the working storage of the computer. Finally, the store command itself has been classified with the other arithmetic commands.

In the discussion of the various computing systems which follows, this group of arithmetic commands will be considered as a block in order to simplify the presentation.

20.8. TRANSFER-OF-CONTROL OPERATIONS

It is convenient to have a group of transfer-of-control operations in any one of the computing systems that will be considered. Therefore, as in the case of arithmetic commands, these will be presented as a group and referred to in a simple manner in the discussion of the various computing systems which follows.

T. Is OP transfer type?, R.
- T.1: TC (Transfer Control): A→ (CC), 0.
- T.2: TZ (Transfer Zero): (AC) = 0?[T.1.], 0.
- T.3: TP (Transfer Positive): (AC) > 0?[T.1.], 0.
- T.4: TN (Transfer Negative): (AC) < 0?[T.1.], 0.
- T.5: TNZ (Transfer Not Zero): (AC) = 0?[0.],[T.1.].
- T.6: TNP (Transfer Not Positive): (AC) > 0?[0.],[T.1.].
- T.7: TNN (Transfer Not Negative): (AC) < 0?[0.],[T.1.].

AC: Accumulator
CC: Command Counter
OP: Operation
R: Return

FIG. 20.15. Sequence scheme for transfer-of-control operations.

In this case the operation may involve simple transfer of control, or transfer on zero, on positive, on negative, on not zero, on nonpositive, and on nonnegative. These various conditions, of course, refer to the contents of the accumulator. Any transfer of control involves transferring the address of the respective commands into the command counter and then going to step 0. Consider statement T.4 of Fig. 20.15, for example. This is a transfer on negative command. A question is asked: Is the number in the accumulator less than zero? If so, go to step T.1, which is a transfer of control; if not, go to step 0, which is the initial step in the general sequence diagram and is the course of events in the case of normal execution of the sequence of commands.

20.9. A SIMPLE TWO-ADDRESS COMPUTER

As indicated in Fig. 20.16, the memory and arithmetic arrangements in the two-address computer are the same as they are in the single-address computer (see Fig. 20.12). The primary difference in structure is in the control unit. Note that there

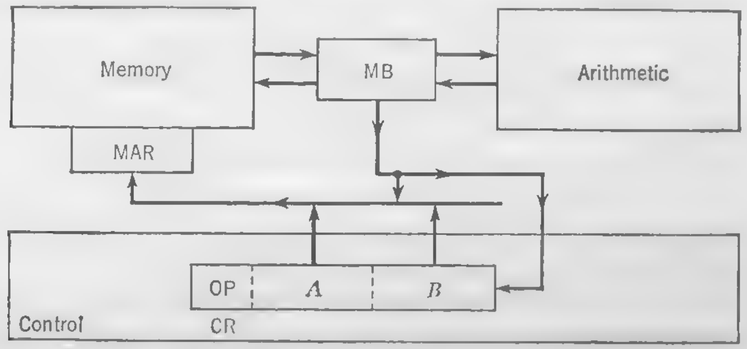


FIG. 20.16. A two-address computer.

is a command register but there is no command counter. The command register is now divided into three parts: the first part stores an operation code, as before, but there are also two addresses labeled A and B. The sequence scheme for the two-address computer is shown in Fig. 20.17. The arrangement here shows one possible

0: B→ (MAR),
1: ((MAR))→ (MB)→ (CR),
2: A→ (MAR),
3: Is OP operand type? ((MAR))→ (MB);
4: OP?
- 4.1. Arithmetic (see Fig. 20.14), 0.
- 4.2. Transfer Control:
  - 4.2.1. TZ (Transfer on Zero): (AC) = 0? A→ (MAR), B→ (MAR); 1.
  - 4.2.2. TN (Transfer Negative): (AC) < 0? A→ (MAR), B→ (MAR); 1.
  - 4.2.3. TP: (AC) > 0? A→ (MAR), B→ (MAR); 1.
  - 4.2.4. TNZ (Transfer Not Zero): (AC) = 0? B→ (MAR), A→ (MAR); 1.
  - 4.2.5. TNP: (AC) > 0? B→ (MAR), A→ (MAR); 1.
  - 4.2.6. TNN: (AC) < 0? B→ (MAR), A→ (MAR); 1.
- 4.3. SUB (Subroutine Transfer): "O"→ (MAR), A→ (MB)→ ((MAR)), B→ (MAR), 1.
- 4.4. RET (Return Transfer): (MB_A)→ (MAR), 1.

FIG. 20.17. Sequence scheme for two-address computer.

way of making use of the second address. The particular type of design described is suitable for computers with cyclic memories. For example, if the information is stored on a magnetic drum the numbers may be requested at a time when that portion of the drum is not under the reading head; consequently, the device must wait until the drum turns to such a position that the information is passing underneath the reading head. In this design the address B specifies the location of the next command, and this removes the need for any command counter.

Referring again to the sequence diagram (Fig. 20.17), step 0 shows that the address B is transferred to the memory access register. In step 1 the command is transferred via the memory buffer to the command register, and in step 2 the address A is transferred to the memory access register. In step 3, if the command is of the operand type, the operand is transferred from the memory to the memory buffer. In step 4 the operation code is interpreted and the arithmetic operation or transfer of control, etc., is carried out.

Step 4.1 describes the various arithmetic operations which are shown in detail in Fig. 20.14. Step 4.2 shows the transfer-of-control operations. These operations must be given in detail since the sequence of the two-address command system is quite different from the sequence of the single-address computer, which is shown in Fig. 20.15. Note that, since each command has the effect of a transfer of control, there is no need for the simple transfer-of-control command. Only the conditional transfer commands, subroutine transfer, and return transfer commands are needed.

The return transfer (step 4.4 of Fig. 20.17) is like the indirect transfer of Fig. 20.13. The number to be transferred into the memory access register does not completely fill the memory buffer. Thus the term MB_A indicates that portion of the memory buffer which corresponds to the address A. This choice of A is consistent with the subroutine transfer command described in step 4.3.

20.10. MULTIPLE-ADDRESS COMPUTERS

In the preceding discussion, any particular command referred to only one operand. In the case of the two-address computer the second address was used for purposes of control or to specify the location of the next command. There is a class of computers with multiple addresses in the command, wherein more than one address refers to operands. Such an example is shown in Fig. 20.18. In actual fact, computers have been made with three addresses as well as the four addresses indicated

here. Note that in this particular design there is still a command counter; so that normally commands will be located in sequence in the memory, and the counting of the command counter determines the actual sequence of execution.

Figure 20.19 shows a sequence scheme for a four-address computer. As in the case of the single-address computer, step 0 consists of incrementation of the command counter. In steps 1 and 2 the command is transferred from the memory into the command register, and in step 3 the OP is decoded.

The addition command (step 3.1) is shown in detail. The first operand, which is specified by the address A, is transferred out to the accumulator; the second operand, specified by address B, is brought out to the memory buffer and added to the contents of A, and the result is left in A. Finally, the third address C is transferred to the memory access register and the result is transferred from the accumulator to the specified location in the memory. In this particular design there is a possibility of

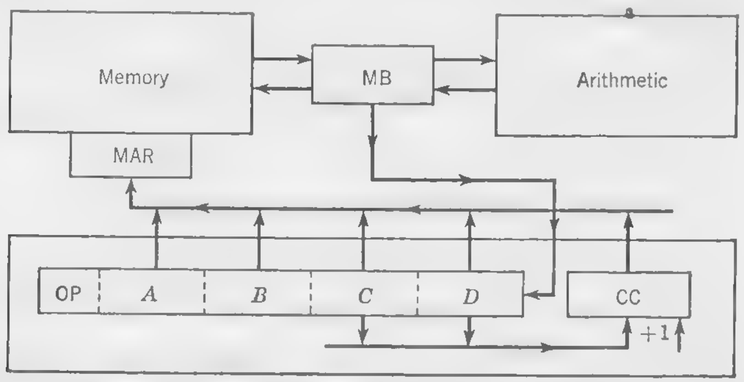


Fig. 20.18. The National Bureau of Standards Western Automatic Computer (SWAC).

overflow. This may happen with either fixed-point or floating-point arithmetic, but in this case assume that the arithmetic is done in fixed-point form and if overflow occurs then the next command will come from a location specified by D rather than as specified by the command counter. Thus each addition command involves a conditional test for overflow, and if overflow occurs there is in effect a transfer of control to a new location. Note that if overflow is to be ignored then the number in D should be the same as the next value of the command counter.

The arithmetic operations for subtracting, multiplying, and dividing are given in a somewhat more concise notation in steps 3.2, 3.3, and 3.4 of Fig. 20.19. Note that in contrast to the single-address computer there is no need for a clear and add, or a clear and subtract, command in this system. In this particular design a number cannot be transferred from one location to another in the memory except by the operation of adding zero to it.

With the multiple addresses the conditional transfer-of-control command becomes essentially a transfer on sign. Depending upon the relative magnitudes of the quantities specified by A and B, the next command may be found in location C, in location D, or in the location specified by the command counter. In this design it becomes feasible to have a transfer and modify command (3.6) which has the effect of modifying an operand and, depending upon the sign of the answer, may or may not transfer control.

Input and output have been ignored in the discussion of the above computers. In step 3.7 a typical input command is shown. The steps are to clear the memory buffer, then to initiate an input process which has the effect of establishing bits which represent the number in the memory buffer. Next, the address C is transferred to the memory access register, and the number is transferred from the memory buffer to the appropriate location in memory. A more complicated version of this input command could be used, in which the A and B addresses would be used to indicate the particular input device, and perhaps to indicate how many numbers were to be read into the

computer. The address D might also be used for a branch in control upon receiving a certain type of "flag" information from the input device.

An output command is shown in step 3.8, in which the address A is transferred to the memory access register, the number in the specified location of memory is transferred to the memory buffer, the output device is started, and the information stored in the memory buffer causes the appropriate information to be recorded on the output medium.

In all computers there is the necessity of being able to get the device started when there is no information stored in the memory of the computer. In the design described by the sequence scheme in Fig. 20.19, this is accomplished by a so-called initial input command (3.9). This command (which is similar to the input command in 3.7, except that the command counter determines the storage of information in the memory

0: (CC) + 1 → (CC),
1: (CC) → (MAR),
2: ((MAR)) → (MB) → (CR),
3: OP?
  3.1. ADD: A → (MAR), ((MAR)) → (AC), B → (MAR), ((MAR)) + (AC) → (AC), C → (MAR), (AC) → ((MAR)), 3.1a: Overflow? [D → (CC), 1.]; 0.
  3.2. SUB: (A) − (B) → (C), 3.1a.
  3.3. MUL: (A) · (B) → (C) (fraction) and (C + 1) (integer), 0.
  3.4. DIV: (A) / (B) → (C) (fraction) and remainder (C + 1), 3.1a.
  3.5. TS (Transfer on Sign): (A) − (B) > 0? [D → (CC), 1.], (A) − (B) = 0? [0.]; C → (CC); 1.
  3.6. TM (Transfer and Modify): (A) − (B) → (C), (C) < 0? [0.]; D → (CC), 1.
  3.7. IN (input): "0" → (MB), Input, C → (MAR), (MB) → ((MAR)), 0.
  3.8. OUT (output): A → (MAR), ((MAR)) → (MB), Output, 0.
  3.9. IIN (initial input): "0" → (MB), Input, (CC) → (MAR), (MB) → ((MAR)), 0.

Fig. 20.19. Sequence scheme for four-address computer.

instead of address C) has the effect of causing information to be read into the computer if the memory of the computer has been cleared. For example, the operation code that represents initial input might be zero. Thus, if the computer's memory has been cleared to zeros, every location stores an initial input command. When the computer is started with zero in the command counter the first command will cause a first word to be read into the computer and stored in location zero. Then the command counter is increased by one, and the next command is again all zeros (coming from location 1), which causes the second word from the input device to be recorded in location 1. This process will continue either until the memory of the computer is completely full or until some other information comes along which terminates the process. An easy way to terminate the process is to have what might be called "flag" information which causes the command counter to be reset. This causes the command in address zero to be executed, and therefore, the routine which has just been read in is executed. The latter might be commands which load other information into the device or execute any desired problem.

20.11. INDEX REGISTERS

One of the important characteristics of a computer is the fact that a relatively small program will process a large amount of information. In order to facilitate this processing of a list of items a concept of index registers has been used. For example, if one wishes to add together two vectors to produce a resultant vector, and the quantities representing the components of these vectors are stored in a sequence of addresses in the memory of the computer, then a single command of the four-address type could add together two components of the vectors and produce a component of the result.

Or; in the case of the single-address computer, three commands could add together two components and produce a component in a result. In order to add together all the components of the vectors and produce a vector answer, it would be necessary to

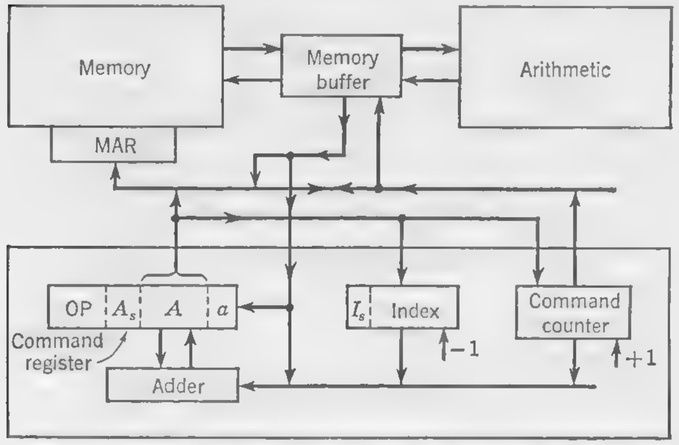


FIG. 20.20. A one-address computer with index register and address modification.

write many such commands or to modify the addresses of the first commands. In order to facilitate this address modification, an index register may be used as shown in the computer described in Figs. 20.20 and 20.21.

0: (CC) + 1 → (CC),
1: (CC) → (MAR),
2: ((MAR)) → (MB), (MB) → (CR),
3: a?
  3.1. a = 0: A → (MB), 5.
  3.2. a = 1: A → (MAR),
  3.3. a = 2: A → (MAR), ((MAR)) → (MAR) → A,
  3.4. a = 3: A + (I) → A, A → (MAR),
  3.5. a = 4: A + (I) → A, A → (MAR), ((MAR)) → (MAR) → A,
  3.6. a = 5: A + (CC) → A, A → (MAR),
  3.7. a = 6: A + (I) + (CC) → A, A → (MAR),
  3.8. a = 7: (AC) → (MB), A + (MB) → A, A → (MAR),
4: Is OP operand type? ((MAR)) → (MB),
5: OP?
  5.1. Arithmetic (see Fig. 20.14).
  5.2. Transfer (see Fig. 20.15).
  5.3. SUB (Subroutine Transfer): 0 → (MAR), (CC) → ((MAR)), A → (CC), 1.
  5.4. IT (Indirect Transfer): (MB) → (CC), 1.
  5.5. LI (Load Index): A → (I), 0.
  5.6. DI (Decrement Index): (I) − 1 → (I), (I) < 0? [0.], [A → (CC), 1.].
  5.7. AM (Add to Memory): (MB) + (AC) → (MB) → ((MAR)), 0.
  5.8. SM (Subtract from Memory): (MB) − (AC) → (MB) → ((MAR)), 0.

A: Address
AC: Accumulator
CC: Command Counter
CR: Command Register
I: Index
MAR: Memory Access Register
MB: Memory Buffer

FIG. 20.21. Sequence scheme for a one-address computer with address modification.

The one-address computer with address modification as shown in Fig. 20.20 is exactly the same as the previous computers with respect to memory and arithmetic circuits. However, the structure of the control unit is somewhat different now. There is an index register along with the command register and the command counter.

The index may be added to the address portion of the command, and to facilitate this operation an adder is shown. However, the index is not always added, so there is an address modifier indicated by the small *a* in the command register which determines whether the index is to be added to the address or not.

The way this index register is used is shown in the sequence scheme of Fig. 20.21.

20.12. THE ADDRESS MODIFIER

In the design described in the sequence scheme of Fig. 20.21 the small *a* in the command register is used to specify other operations rather than just the addition of the index to the address A. In this example the *a* determines the number of references to the memory and provides the right operand. Also, it provides for modification of the address by the contents of the command counter or by the contents of the accumulator, as well as by the index register. The use of these various facilities will be indicated by examples in what follows.

a	Operand
0	A
1	(A)
2	((A))
3	(A + (I))
4	((A + (I)))
5	(A + (CC))
6	(A + (CC) + (I))
7	(A + (AC))

FIG. 20.22. Operand table for the one-address computer with address modification.

Note that the address and the index register both carry signs and that when the address is transferred to the memory access register the sign of A has no effect. Thus the sign of A is important with respect to the addition operation but does not change the location of the corresponding number in memory.

The operands which are prepared by steps 3 and 4 of Fig. 20.21 are shown in detail in Fig. 20.22. For example, if small *a* is equal to zero, the operand is the address A itself. In this case it may be assumed that the sign of A is transferred to the memory buffer in the proper position so that the number A may be a positive or negative integer within the range of A. Note that this happens in step 3.1 where A is transferred directly to the memory buffer. The next step is step 5 where the operation code is decoded.

If *a* is equal to 1 there is a reference to the memory for the operand and this is similar to the situation in the computers described above.

If *a* is equal to 2, then the address is transferred to the memory access register and the content of the corresponding location of memory is transferred to the memory buffer (not shown in Fig. 20.21) and to the memory access register. This produces indirect addressing so that the final number referred to comes from a location determined by the first reference to the memory. Such commands are useful in processing items in lists, and particularly in lists of items where the item does not fit directly into the word length of the respective computer.

If *a* is equal to 3, then the address is modified by the contents of the index register and this result is used as the address of the operand.

If *a* is equal to 4, then the same thing happens except that there is one extra reference to the memory; so that this is again an example of indexed indirect addressing.

If *a* is equal to 5 or 6, the contents of the command counter are added to the address before the reference to memory occurs. This makes it easy to write subroutines which are relocatable in memory without modification of commands.

Example. A subroutine may be written as if it starts in address zero, but any references to locations within the subroutine will have small *a* equal to 5 or 6. This causes the contents of the command counter to be added to the respective addresses

before execution. If the subroutine starts at address 1000, then any operand will be found in the correct location, or transfers within the subroutine will be to the correct locations.

If a is equal to 7, the contents of the accumulator are transferred to the memory buffer, then added to the address A, and the result is used as an operand. This particular type of command is useful in complicated subscript operations. For example, a complicated subscript may need to be computed in the accumulator, and then added to an address to provide the final reference to the memory. This all can happen in a single command by making use of statement 3.8. Note that if the arithmetic is done in so-called normalized floating-point form then the transfer of the contents of the accumulator to the memory buffer may need to be preceded by a shift to zero exponent or to integral form, wherein the significant figures of the number are an integer in the address position of the word.

A list of operation codes is given in step 5 of Fig. 20.21. The arithmetic and transfer operation codes are as have been described in Figs. 20.14 and 20.15. The subroutine and indirect transfers have also been described before. With the addition of the index register, load index and decrement index commands have been added. The load index (5.5) transfers the resultant address into the index register. The decrement index (5.6) has the effect of subtracting 1 from the contents of the index register, and tests to see if the index is negative. If the index is negative this corresponds to a completion of the cycle, and the sequence scheme goes to step 0, causing the next following command to be executed. If the result is nonnegative, then there is a transfer of control, presumably back to the initial command of the loop.

It is interesting to note that index registers were first used in a computer designed at Manchester University in England, and that the first commercial computer to use an index register in the United States was the Datatron, which was then manufactured by the Consolidated Engineering Corporation.

As an indication of possible variations in arithmetic commands, an add to memory and subtract from memory are shown in steps 5.7 and 5.8 of Fig. 20.21.

20.13. A COMPUTER WITH PAGE MEMORY AND INTERRUPT LOGIC

Some of the features described above plus other features now appearing in currently designed computers have been gathered together and presented in the computer described in Figs. 20.23 and 20.24. Inspection of Fig. 20.23 shows that the general arrangements of the memory and arithmetic circuits are still the same; in the control circuits there are still a command register and command counter but there are also other registers. These are called the "page register," the "address register," and the "flag register." There are in addition some other processing facilities, for example, a complementing unit labeled K, an adding circuit, and an unequal detection circuit associated with the page register.

The command register now has four parts: an operation code, an address A, an address modifier a, and an index designator I. In contrast to the design discussed just above, the address modification and the specification of the index register appear as separate items in the command.

The various counters, such as the command counter and address registers, are no longer than necessary to specify the number of locations in the high-speed memory. This is particularly obvious in considering the memory access register which consists of three parts; the part labeled MAR-3 corresponds to the old memory access register and will store an integer of sufficient size to specify any one of the locations in the high-speed memory.

In this computer it will be assumed that there is another slower-speed and larger memory available to the system. The information from this auxiliary memory may arrive via the arithmetic circuits or the memory buffer or some special input-output buffer register. In this discussion the control of the arrival, and transmission of information to such a memory, and the manner in which the programmer refers to that information, are all that will be discussed. The fast memory will be considered to be divided into pages labeled M1, M2, M3, and M4, and the portion of the memory access register labeled MAR-2 will specify which particular page is being considered. Also,

the auxiliary memory will be considered to be divided into pages, and the specification of the particular auxiliary memory page will be given by MAR-1.

Before considering in detail the operation of the page register, it is of interest to consider the operation of the flag register, which is described in step 0 of the sequence scheme of Fig. 20.24. The flag register is like a flagman in a railway system. It may call a halt to the current operation and cause some other action to take place in the meantime. The actual character of the interruption is determined by a portion of the flag register labeled F, and the binary bits in this portion of the flag register may be associated with various devices which have the ability to "flag down" the computer. For example, a typewriter may have the ability of setting the first bit in this flag

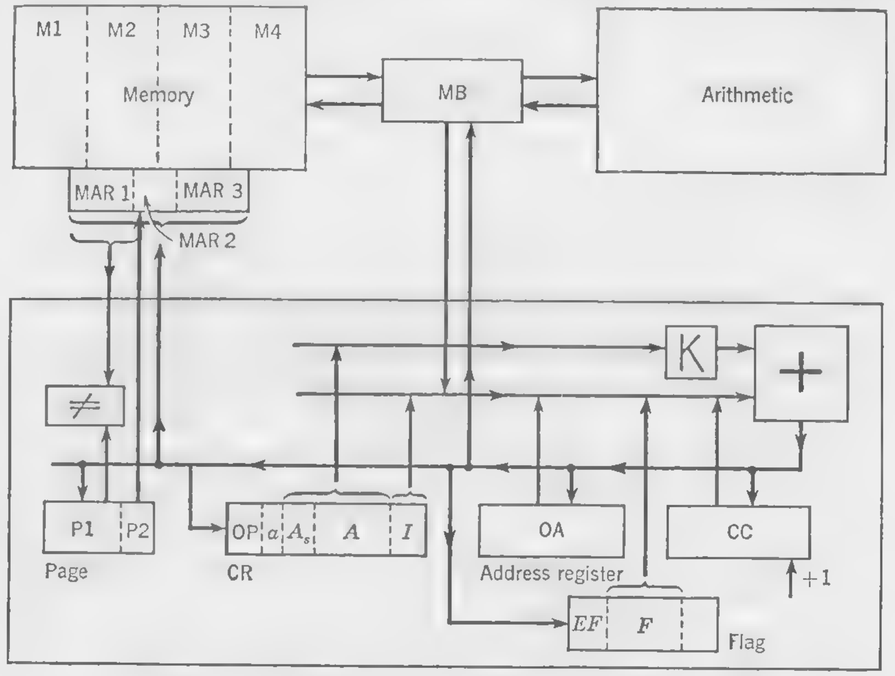


FIG. 20.23. Computer with page register and interrupt logic.

register F, or a magnetic-tape unit may be associated with the second bit, F2, etc. Thus, while a program is in operation in the computer, someone may hit a key upon the typewriter, causing an interrupt bit to be established in the flag register F. The next time that the computer moves through step 0 of the sequence scheme there will be an inspection of the flag register and the enable flag EF. If both these quantities are different from 0, an interruption will take place, which is characterized by a transfer of 0 to the memory access register and a storing of the contents of the command counter in location 0 of the memory. This is followed by a transfer of control to a location which depends upon the combination of flags F. Note that this operation is very similar to a subroutine transfer (see step 4.4 of Fig. 20.24). Note also that, if no flags are present, or if the enable flag has not been set, the command counter is incremented and the device continues to execute commands in sequence.

In the modification of addresses, the index register is considered separately from the other possibilities of modification. In step 2 if an index is present that index specification is transferred to the memory access register, and the number in the operand address register (which is the current value of the address of the specified command) has a number from memory added to it. Thus in this case, the I specifies the location in memory of the corresponding index register. One may have as many index registers as one desires simply by making the command register, or more specifically, the I portion of the command register, as long as one pleases. In fact, if I is the same size as the address A, any location in the fast memory may be used as an index. As a practical matter it is probably reasonable to limit I to 63 locations in memory.

In this particular design there are four possible values for the modifier a.

In order to simplify the presentation, all the details of picking up an operand have not been shown. For example, in step 3 an address is developed in the operand address register. Statements 4.1, in particular, expect an actual operand to have been developed in the memory buffer. The details of this operation are easy to supply.

With a equal to 0 or equal to 1, the operand address is increased by either A or the number in memory specified by A. If a is equal to 2, the operand address is increased

```
0: EF ∧ F? ["0" → (MAR), (CC) → ((MAR)), "0" → EF, F → (CC)], (CC) + 1 → (CC);
1: (CC) → (MAR), ((MAR)) → (CR),
2: I? [I → (MAR), (OA) + ((MAR)) → (OA)];
3: a?
  3.1. a = 0: (OA) + A → (OA) → (MB).
  3.2. a = 1: A → (MAR), (OA) + ((MAR)) → (OA) → (MB).
  3.3. a = 2: (OA) + A → (MAR), ((MAR)) → (OA) → (MB).
  3.4. a = 3: (OA) + (CC) + A → (MAR), ((MAR)) → (OA) → (MB).
4: OP?
  4.1. Arithmetic (see Fig. 20.14), 5.
  4.2. Transfer (see Fig. 20.15), 5.
  4.3. NOP (No operation): 0.
  4.4. SUB (Subroutine): (CC) → (MB), "0" → (MAR), (MB) → ((MAR)),
                         (OA) → (CC), 1.
5: "0" → (OA), 0.
```

FIG. 20.24. Sequence scheme for a computer with interrupt.

by A, and there is a reference to memory for the corresponding result. Finally, if a is equal to 3 the contents of the command counter are added to the operand address and a reference to memory is made. This provides for the possibility of relocatable subroutines without actual address modification.

Note that the operation code of step 4.3 is called "no operation." This does not clear the contents of the OA register (see step 5). Consequently, a number of commands may have the purpose of building up a complex operand or address in the operand address register.

The subroutine transfer of step 4.4 is similar to those discussed before. Note that there is no need for a return command in this system, since the address-modification facilities of step 3 already provide for indirect addressing in the case of transfers of control. Note that here, as in the last example, the address A carries a sign digit which is not used in transferring information into the MAR-3 register.

Suppose that the fast memory shown in the upper left-hand corner of Fig. 20.23 stores 4,000 words, numbered from 0 to 3,999. Suppose further that the command counter has stepped up through the value 3,999, and therefore it goes on to the value of 4,000. When this number is transferred via the control circuits to the memory access register, some bits of information appear in the MAR-1 portion of the memory access register. Under normal circumstances these bits do not compare with what appears in the page register in the portion P1.

```
((MAR)) → (MB) may be replaced by:
  [(MAR-1) = 0? ((MAR-3)) → (MB),
    [(MAR-1) = (P1)? [(P2) → (MAR-2), ((MAR-3)) → (MB)],
                     ["1" → (F1), 0.]; ].
```

FIG. 20.25. Homogeneous memory system with page registers.

As indicated in Fig. 20.25, called "homogeneous memory system with page registers," the content of MAR-1 is compared with 0, or with the contents of P1. If neither of these comparisons succeeds, a 1 is transferred to the flag register at F1, and there is a return to step 0, which normally causes an interruption. When the interruption occurs, an executive routine comes into operation which (a) inspects the contents of the P1 portion of the page register, (b) causes a search in the auxiliary memory, and (c) brings in a new page and stores it in appropriate location in the M1, M2, M3, or M4 parts of the memory.

The first time that a page is referred to that is not in the high-speed memory there will be a search through the auxiliary memory for this page, which may take a relatively large amount of time. However, subsequent references to that page will be made at high speed.

It is an interesting problem to develop an algorithm which decides which one of the blocks M1, M2, M3, or M4 is replaced when a new page is read into the high-speed memory. It is assumed in this case that page M1 will never be replaced in order that the executive routine will always be available to process requests for new pages. [Whenever a new page is established in the memory, the number of that page is established first in the P1 portion of the page register, and the appropriate designation of M2, M3, or M4 is established in P2. If there is a subsequent reference to this page then (P2) is transferred to MAR-2 in order that the reference to the proper portion of the high-speed memory will be made.] One algorithm for deciding which page, M2, M3, or M4, to replace, says to choose this at random. Other more elaborate algorithms which inspect the use of the respective pages may be used, but the optimum selection certainly depends upon the type of problem and the details of the structure of the auxiliary memory of the computing system.

A more complete system might have a number of page registers, and the high-speed memory might be divided into a larger number of pages.

The features described above, such as the addition of the command counter to the address of the command and the page-register feature, are expected to reduce the labor of preparing programs for such computers. In many installations, however, compiling systems have been developed which automatically take care of these features, even without the specific hardware facilities being present. In such systems the additional hardware has the effect of simplifying the compiling routine and is presumably justified by the improvement in efficiency in the over-all computing system.

References

1. Alrich, John C.: Engineering Description of the ElectroData Digital Computer, *IRE Trans. on Electronic Computers*, **EC-4**(1): 1–10 (March, 1955).
2. Bauer, Walter F.: An Integrated Computation System for the ERA-1103, *J. Assoc. Computing Machinery*, **3**(3): 181–185 (July, 1956).
3. Bazilevskii, IU. IA: The Universal Electronic Digital Machine (URAL) for Engineering Research, *J. Assoc. Computing Machinery*, **4**(4): 511–519 (October, 1957).
4. Beard, A. D., L. S. Bensky, D. L. Nettleton, and G. E. Poorte: Logic Design of the RCA BIZMAC Computer, *IRE Convention Record*, part 4, New York, 1956, pp. 81–87.
5. Block, Niel: The Alwac Corporation Model 800 Computer, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 118–124.
6. Blumenthal, Sherman: A Dual Master File System for a Tape Processing Computer, *J. Assoc. Computing Machinery*, **5**(4): 319–327 (October, 1958).
7. Bracken, Robert H., and Bruce G. Oldfield: A General System for Handling Alphameric Information on the IBM 701 Computer, *J. Assoc. Computing Machinery*, **3**(3): 175–180 (July, 1956).
8. Buchholz, W.: Design Objectives for the IBM Stretch Computer, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 99–105.
9. Carter, W. C.: A New Large-scale Data Handling System, DATAmatic 1000, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 36–56.
10. Churchill, Alex B.: A Small, Low-cost Business Computer, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 187–189.
11. Dean, Franklin R.: Operating Experience with RAYDAC, *Review of Input and Output Equipment Used in Computing Systems—Joint AIEE-IRE-ACM Computer Conference*, New York, S53, Dec. 10–12, 1952, pp. 77–79.
12. Dover, Jerome J.: Centralized Data Processing System, *Trends in Computers: Automatic Control and Data Processing*, *Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 172–183.
13. Dreyfus, P. L., H. G. Feissel, and B. M. Leclerc: A Magnetic Drum Extension to the Gamma 3 Computer, *IRE Convention Record*, part 4, New York, 1956, pp. 105–108.
14. Dunwell, S. W.: Design Objectives for the IBM Stretch Computer, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 20–21.
15. Eckert, J. P.: Univac-Larc, the Next Step in Computer Design, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 16–19.
16. Everett, R. R., C. A. Zraket, and H. D. Benington: SAGE—A Data-processing System for Air Defense, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 148–155.

17. Frankel, Stanley P.: The Logical Design of a Simple General Purpose Computer, *IRE Trans. on Electronic Computers*, **EC-6**(1): 5–14 (March, 1957).
18. Gehring, A. J., L. W. Stowe, and L. D. Wilson: The Univac Magnetic Computer—Part I. Logical Design and Specifications (Abstract), *IRE Convention Record*, part 4, New York, 1956, p. 109.
19. Githens, J. A.: The Tradic Leprechaun Computer, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 29–33.
20. Greenstadt, J. L.: The IBM 709 Computer, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 92–96.
21. Hughes, E. S., Jr.: The IBM Magnetic Drum Calculator Type 650—Engineering and Design Considerations, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 140–154.
22. Huskey, Harry D., and David C. Evans: The Bendix G-15 General Purpose Computer, *Proc. WESCON Computer Sessions*, Los Angeles, Calif., Aug. 25–27, 1954, pp. 87–91.
23. Kranzley, A. S., *et al.*: BIZMAC II Computer, Characteristics and Application, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 57–69.
24. Lebedev, S. A.: The high-speed Electronic Calculating Machine of the Academy of Sciences of the U.S.S.R., *J. Assoc. Computing Machinery*, **3**(3): 129–133 (July, 1956).
25. Leiner, A. L.: System Specifications for the DYSEAC, *J. Assoc. Computing Machinery*, **1**(2): 57–81 (April, 1954).
26. Leiner, A. L., and S. N. Alexander: System Organization of the DYSEAC, *IRE Trans. on Electronic Computers*, **EC-3**(1): 1–10 (March, 1954).
27. Leiner, A. L., W. A. Notz, J. L. Smith, and A. Weinberger: System Design of the SEAC and DYSEAC, *IRE Trans. on Electronic Computers*, **EC-3**(2): 8–23 (June, 1954).
28. Lesser, M. L., and J. W. Haanstra: The RAMAC Data-processing Machine, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 139–146.
29. Lipton, S.: A Note on the Electronic Computer at Rothamsted, *Mathematical Tables and Other Aids to Computation*, **IX**(50): 69–70 (April, 1955).
30. MacDonald, D. N.: Datafile—A New Tool for Extensive File Storage, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 124–127.
31. Maddox, J. L., J. B. O'Toole, and S. Y. Wong: The Transac S-1000 Computer, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 13–16.
32. Merwin, Richard E.: The IBM 705 EDPM Memory System, *IRE Trans. on Electronic Computers*, **EC-5**(4): 219–223 (December, 1956).
33. Mitchell, J. L., and K. H. Olsen: TX-O, A Transistor Computer, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 93–101.
34. Morton, Paul L.: The California Digital Computer, *Mathematical Tables and Other Aids to Computation*, **V**(34): 57–61 (April, 1951).
35. Perkins, Robert: EASIAC, A Pseudo-computer, *J. Assoc. Computing Machinery*, **3**(2): 65–72 (April, 1956).
36. Porter, V. J., S. E. Smith, and M. Naiman: The Univac Magnetic Computer—Part III. Drum Memory (Abstract), *IRE Convention Record*, part 4, New York, 1956, p. 111.
37. Rosenberg, Jack: Logical Organization of the DIGIMATIC Computer, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 25–29.
38. Shiowitz, M., A. A. Cherin, and M. J. Mendelson: Functional Description of the NCR 304, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 34–38.
39. Smith, B. K.: The Univac Magnetic Computer—Part II. Megacycle Magnetic Modules (Abstract), *IRE Convention Record*, part 4, New York, 1956, p. 110.
40. Smith, J. Ernest: A New Large-scale Data-handling System, DATAmatic 1000, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 22–28.
41. Sumner, J. S.: Magnetic Tape File Processing with the NCR 304, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 9–17.
42. Vance, P. R., L. G. Dooley, and C. E. Diss: Operation of the SAGE Duplex Computers, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 160–163.
43. Wall, D. D.: Multiplication Time on the IBM 709, *Mathematical Tables and Other Aids to Computation*, **XII**(63): 217–218 (July, 1958).
44. Welsh, H. F., and V. J. Porter: A Large-capacity Drum-file Memory System, *Proc. Eastern Joint Computer Conf.*, T-92, New York, Dec. 10–12, 1956, pp. 136–138.
45. Withington, Frederic G.: The Cardatron and the Datafile in the Datron System, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 19–33.
46. Wolf, J. Jay: The Office of Naval Research Relay Computer, *Mathematical Tables and Other Aids to Computation*, **VI**(40): 207–212 (October, 1952).
47. Wong, S. Y.: Philco S-2000 Transistorized Large-scale Data Processing System, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 106–117.
48. Zimmer, E. D.: The X308 Computer, *New Computers—A Report from the Manufacturers*, Los Angeles, March, 1957, pp. 72–90.

Section 21

APPLICATIONS OF DIGITAL COMPUTERS

By

CHARLES W. ADAMS, *President, Charles W. Adams Associates, Inc., Bedford, Mass.*

RICHARD G. CANNING, *Consultant, Vista, Calif.*

HARRY D. HUSKEY, *Professor of Mathematics and Electrical Engineering, University of California, Berkeley, Calif.*

ARVID W. JACOBSON, *Professor, Department of Mathematics, Wayne State University, Detroit, Mich.*

E. CALVIN JOHNSON, *Head, Computer Development Department, Research Laboratories Division, The Bendix Corporation, Detroit, Mich.*

SAUL ROSEN, *Consultant, Computing and Programming Systems, Philadelphia, Pa.*

MORRIS RUBINOFF, *Professor, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pa.*

ROGER SISSON, *Aeronutronic, Ford Motor Company, Newport Beach, Calif.*

JAMES H. WILKINSON, *Mathematics Division, National Physical Laboratory, Teddington, Middlesex, England.*

CONTENTS

21.1. Scientific and Engineering Applications of Digital Computers 21–2
21.1.1. Introduction 21–2
21.1.2. Solution of Linear Systems and Matrix Computations 21–2
21.1.3. Ordinary Differential Equations 21–4
21.1.4. Two Boundary Problems, Including Eigenvalue Problems 21–6
21.1.5. Integro-differential Equations 21–7
21.1.6. Nonlinear Algebraic Equations 21–8
21.1.7. Partial Differential Equations 21–8
21.1.8. The Use of An Electronic Computer in Testing the Design of a Proton Synchrotron 21–10
21.1.9. Crystallographic Calculations 21–11
21.1.10. Analysis of Stochastic Processes 21–11
21.1.11. Data Reduction 21–11

21.1.12. Specific Areas of Application..... 21–11
21.2. Data Processing..... 21–13
21.2.1. Introduction..... 21–13
21.2.2. Businesses and Their Data-processing Problems 21–13
21.2.3. Characteristics of Business Data Processing.... 21–17
21.2.4. Multitape Files Using Alternate Tape Units..... 21–19
21.2.5. Unit Records, Fields, Keys, and Trailers..... 21–19
21.2.6. Factors Determining the Magnitude of a Problem. 21–21
21.2.7. Criteria for Evaluating Equipment..... 21–22
21.2.8. The Need for Sorting.... 21–22
21.2.9. Sorting Techniques..... 21–24
21.2.10. Choice of Sorting Techniques..... 21–32
21.3. Management Problems..... 21–33
21.3.1. The Problem Defined.... 21–33
21.3.2. The State of the Art..... 21–34
21.3.3. Economic Considerations 21–40
21.4. Simulation..... 21–42
21.4.1. Principles of Simulation. 21–42
21.4.2. Scope of the Section on Simulation..... 21–43
21.4.3. Function Approximation. 21–44
21.4.4. Dynamic Behavior of Digital Simulators..... 21–49
21.4.5. Mixed Digital-analog Simulation..... 21–54
21.4.6. Examples of Digital Simulation..... 21–56
21.5. Computers and Control..... 21–62
21.5.1. Machine-tool Controls... 21–63
21.5.2. Equation-solving Controls..... 21–81
21.5.3. Digital-operational Techniques..... 21–89

21.1. SCIENTIFIC AND ENGINEERING APPLICATIONS OF DIGITAL COMPUTERS*

21.1.1. Introduction. Digital computers have been used very extensively in the solution of engineering and scientific problems. Only a brief sketch of some of the major fields of application can be included here. The need for large-scale scientific computation provided the major impetus for the development of electronic computers, and computer characteristics were determined to a large extent by the nature of the applications. Thus Eniac and Edvac (University of Pennsylvania) were designed in response to the computing needs of the Ballistics Research Laboratory at Aberdeen, Md. Whirlwind (MIT) was designed at least in part for the handling of real-time control problems. The SWAC and SEAC (National Bureau of Standards), the EDSAC (Cambridge University, England), the Manchester (England) Computer, and the Pilot ACE (N.P.L., England) were all designed for general scientific calculation. The Institute for Advanced Study computer and its successors, the Ordvac, Illiac, Oracle, etc., were designed in connection with weather-forecasting and atomic-research problems. It was largely the computing needs of the aircraft industry that led to the development of the present-day large-scale electronic computers. Problems in partial differential equations arising mainly in the application of nuclear physics provided the major motivation for the new ultra-high-speed computers.

21.1.2. Solution of Linear Systems and Matrix Computations. Let there be given the n linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{21.1}$$

The values of the unknowns $x_1, x_2, \ldots, x_n$ are to be determined. This problem arises in a great number of contexts. In matrix notation (21.1) may be written $Ax = b$ and the solution in the form $x = A^{-1}b$ where A^{-1} is the inverse of A.

Matrix methods have been used very extensively on high-speed computers on a wide range of problems. The fundamental programs are those dealing with matrix

* By Arvid W. Jacobson, Saul Rosen, and James H. Wilkinson

multiplication, the solution of linear algebraic equations including matrix inversion, the evaluation of determinants, and the calculation of the latent roots and vectors of matrices. Apart from being used on problems which have presented themselves directly in matrix form, the methods have found extensive application to the solution of differential equations, integro-differential equations, and eigenvalue problems associated with systems of differential equations. Matrices arising from such systems frequently have a high percentage of zero elements, and because they may be of very high orders, programs have been prepared which take advantage of the presence of zero elements.

In Great Britain, a panel was established whose aim has been to devise a number of standard forms in which matrix problems should be framed whenever possible, in order to minimize the amount of programming needed. The reason for the prevalence of matrix forms is that for the purposes of analysis an aircraft is frequently represented by a number of point masses with rotational inertias and elastic interconnections. With the use of electronic computers it has been practicable to use many more point masses than hitherto. In calculations associated with flutter two main problems of general mathematical interest predominate. The first arises in the calculation of the normal modes of the aircraft. This requires the calculation of the latent roots and vectors of an unsymmetric matrix. Matrices of orders higher than 60 have been dealt with, usually only the higher roots and their vectors (of the order of 10) being required. The matrix D of which the latent roots are required is itself defined by the matrix relation

$$D = [I - C(C'MC)^{-1}C'M]\,SM$$

and its computation requires eight matrix multiplications. The second main problem is the determination of the critical speeds with respect to flutter. The equations are of the form

$$A\ddot{x} + B\dot{x} + Cx = 0$$

where A, B, and C are $(n \times n)$ matrices and x is a vector with n components. C is a function of the velocity V. The calculation of the matrices A, B, and C requires a considerable number of matrix operations. The solution of this set of equations is of the form

$$x = \Sigma\alpha_i e^{\lambda_i t}$$

where in general the λ_i are complex. The motion is stable for the values of V for which all the λ_i have negative real parts. A critical value of V is one for which one of the λ_i takes a pure imaginary value, and such a value separates the velocities for which the motion is stable from those for which it is unstable. Besides requiring the critical values of V, the values of λ_i at speeds in the neighborhood of a critical value are needed since these give the damping factors. By a simple matrix manipulation the problem can be reduced to that of finding the complex roots of a real matrix for each value of V. The order of the matrix is double the number of dependent variables in the differential equation. The problem is of general interest since the motions of many damped mechanical systems about positions of equilibrium are governed by systems of differential equations of this type.

The following are a few situations leading to linear systems:

Analysis of Structures. Forces at various points in a structure resulting from given applied forces are to be determined. This problem arises in the design of airplanes, automobiles, bridges, and other large structures.

Electrical-network Problems. The currents or voltages at various points in a circuit are to be determined when certain applied voltages and current sources are given. In this application the coefficients and the solutions are in general complex numbers. The computer may be programmed to do complex arithmetic, or the problem may be reduced to real arithmetic. Usually the latter alternative involves doubling the size of the system of equations that must be solved.

Curve Fitting. The problem of fitting a set of empirical data by the best possible polynomial of given degree can be reduced to the solution of a linear system provided

that best possible fit is defined in the sense of least squares. Other curve-fitting techniques may also lead to linear systems.

Multiple Correlation. This statistical technique is equivalent to fitting a plane to empirical data by the method of least squares and is carried out by solving a linear system. The inverse of the covariance matrix yields the various partial correlation coefficients.

Difference Equations. Problems in heat distribution, in electromagnetic-field theory, in neutron diffusion, and in numerous other contexts give rise to boundary-value problems involving linear differential equations, both ordinary and partial. One standard method for solving such problems is to replace differential equations by approximating difference equations. The boundary-value problem is then replaced by a system of linear algebraic equations.

21.1.3. Ordinary Differential Equations. *Initial-value Problems.* Problems in mathematical physics and engineering arise more frequently in this form than in any other. An interesting feature of the work on high-speed computers is that the method of solution which has been used most frequently is one which is seldom used on desk computing machines. It is an extension devised by S. Gill of a method due to Runge and Kutta and usually known as the modified Runge-Kutta process. In this method the system of differential equations is expressed as a set of simultaneous first-order differential equations of the form

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, \ldots, y_n) \qquad i = 1, 2, \ldots, n$$

For convenience in the program the extra equation

$$\frac{dx}{dx} = 1$$

is usually introduced. Equations containing higher derivatives are expressed in this form by the introduction of extra dependent variables. Thus, if an equation contains the third derivative of y_1, extra variables y_2 and y_3 are introduced defined by

$$\frac{dy_1}{dx} = y_2 \qquad \text{and} \qquad \frac{dy_2}{dx} = y_3$$

d^3y_1/dx^3 may then be replaced by dy_3/dx. The integration proceeds step by step, values of the dependent variables being obtained for a sequence of values of the independent variable. The method differs from those employing differences which are usually used on desk machines, in that in each step of the integration no use is made of values obtained at previous steps. This means that storage requirements are minimized, no special treatment of the first step is required, and there are no special problems associated with the change of the interval of integration. The main weaknesses of the method are:

1. It is sometimes slow, in that it may be necessary to perform the integration at smaller steps than would be necessary in a method in which high-order differences were used.

2. There is no convenient way in general of estimating whether the step length is small enough for the approximations involved in the method to be sufficiently accurate to meet requirements.

The first of these objections is of little importance when the values of the dependent variable are needed at an interval roughly equal to that which may be used in the modified Runge-Kutta process or when the speed of output of results rather than that of computation is the limiting factor. A decision on the optimum interval is usually made on the basis of a few trial runs, and if the problem contains a number of parameters so that in all many integrations are required, these trial runs do not add significantly to the volume of work. Because the programming requirements are reduced

to a minimum when using the modified Runge-Kutta process, it has proved worthwhile to use an electronic computer on quite simple sets of differential equations where it might otherwise have appeared uneconomic as compared with desk machines. Typical examples where it has been used are given below.

As far as sheer bulk is concerned, the greatest number of differential-equation systems which have been solved have been associated with calculating the paths of various projectiles including bombs, shells, and guided missiles of many types. For the former the differential equations are of the type

$$\ddot{x} = -Cv\dot{x}g(v)f(y)$$
$$\ddot{y} = -g - Cv\dot{y}g(v)f(y)$$

where x and y are the coordinates of the shell, v is the velocity, C is a constant depending on the type of shell, $g(v)$ is an empirical function of the velocity, and $f(y)$ varies as the density of the atmosphere at height y. The empirical function $g(v)$ is computed by the machine at each point of the integration, using polynomial or other approximations. Frequently the machine itself has been used to determine these approximations to the empirical function. Usually integrations have been associated with the production of range tables, but occasionally the object has been to assess the merit of an empirical approximation to $g(v)$. Usually the system of equations used in guided-missile calculations has been of much higher order than that of the set above, and frequently an important feature has been the presence of random influences on the path of the projectile due to noise in the radar information. These random influences, which have to satisfy appropriate statistical requirements, have usually been generated by a program on the computer simultaneously with the solution of the differential equations. Because of the random element, the behavior of the differences of the dependent variables is erratic, so that the Runge-Kutta method is more suitable than methods based on differences.

An example of a very simple set of nonlinear differential equations for which a completely theoretical investigation was not found practical arose in the theory of a self-exciting generator. It concerned a system consisting of a disk rotating on an axis with a coil touching the axis and rubbing on the periphery of the disk and shunted by a second coil with finite resistance.

The equations governing the system are of the form

$$\frac{dI}{dt} + aI = wI$$
$$\frac{dI_1}{dt} + bI_1 = wI$$
$$\frac{dw}{dt} = 1\text{-}I(I + I_1)$$

For some value of a and b the equilibrium position is unstable and it was found that I and I_1 oscillate unboundedly as t tends to infinity, in such a way that they spend more and more of the time at near-zero values but have occasional peaks which become higher and narrower with increasing t.

Another example is the problem of calculating the efficiency of capture of small particles by spherical drops in a fluid medium. Two cases were treated, that of potential flow and that of viscous flow. When reduced to dimensionless form the equations for the motion of the particles become

For potential flow:

$$\lambda \frac{d^2x}{dt^2} = 1 - \frac{x^2 - (y^2/2)}{(x^2 + y^2)^{5/2}} - \frac{dx}{dt}$$

and

$$\lambda \frac{d^2y}{dt^2} = -\frac{3xy}{2(x^2 + y^2)^{5/2}} - \frac{dy}{dt}$$

For viscous flow:

$$\lambda \frac{d^2x}{dt^2} = 1 - \frac{1}{(x^2+y^2)^{1/2}} - \frac{x^2+y^2-1}{2}\frac{2x^2-y^2}{2(x^2+y^2)^{5/2}} - \frac{dx}{dt}$$

and

$$\lambda \frac{d^2y}{dt^2} = -\frac{x^2+y^2-1}{2}\frac{3xy}{2(x^2+y^2)^{5/2}} - \frac{dy}{dt}$$

The paths of primary interest were those which just grazed the sphere. Trajectories were calculated for a considerable range of values of λ.

One of the earliest problems in this class to be treated on an electronic computer was that of the second-order corrections to measurements of gravity made at sea, which required the analysis of the motion of a pendulum supported on a periodically accelerated support. The computation was carried out before the modified Runge-Kutta method had been developed. The equations were solved for two main cases, (1) for a single horizontal acceleration and (2) for a single vertical acceleration.

(1) involved, first, the solution of

$$\theta'' = -\left(\frac{g}{l}\right)\theta + \frac{F}{l}\sin(\omega t + P)$$

with initial conditions

$$\theta(0) = F\sin\frac{P}{g} + A \qquad \theta'(0) = \omega F\cos\frac{P}{g}$$

to give θ_d, and with the initial conditions

$$\theta(0) = F\sin\frac{P}{g} \qquad \theta'(0) = \omega F\cos\frac{P}{g}$$

to give θ_r, followed by the solution of the equation

$$E'' = -\left(\frac{g}{l}\right)E + \frac{AF}{2l}(\theta_d + \theta_r)\cos\left[\left(\frac{g}{l}\right)^{1/2}t\right]\sin(\omega t + P)$$

to give the correction E to be applied to the recorded motion to produce the motion of a single, undisturbed, damped cycloidal pendulum.

The equations for (2) were of a very similar nature.

The following example, involving the calculation of special perturbations, is an example of a problem in which the Runge-Kutta process compared very unfavorably with conventional methods taking into account high-order differences. The problem was the calculation of the paths of minor planets taking into account the attraction of the seven major planets. The equations to be integrated are the Newtonian equations of motion. If the modified Runge-Kutta process had been used a step length of the order of 5 to 10 days would have been needed and a feature of the process is that values are needed at three intermediate points for each step. Since the equations of motion involve the positions of the perturbing bodies which must be given from tables, it was important to keep to a minimum the number of points at which the equations of motion were applied. Using a more conventional method of integration in which sixth-order terms were taken into account, a step length of the order of 20 days was permissible.

21.1.4. Two Boundary Problems, Including Eigenvalue Problems. When the boundary values are not all prescribed at one end of the range of integration, the method of forward integration has still sometimes been used. For a second-order differential equation in which the value of the dependent variable is given at both ends of the range, integration can proceed from one end of the range with an assumed value of the derivative. The correct initial value of the derivative may be found by some form of interpolation in the values obtained for the dependent variable at the other end of the range for different initial values of the derivative. When an eigenvalue problem is involved similar methods have still been used successfully in some circumstances. A typical problem is to find the value of λ for which a nonzero solution exists of the equation

$$f(x)\frac{d^2y}{dx^2} + g(x)\frac{dy}{dx} + [L(x) + \lambda]\,y = 0$$

with the boundary conditions

$$y = 0 \qquad \text{when } x = a \text{ and when } x = b$$

The corresponding solution is known as the eigenfunction. Examples of such equations arise in atomic theory, the simplest case being that for the behavior of one electron under a simple central field of force. The equation is of the form

$$\frac{d^2P}{dr^2} + \left\{2[E - V(r)] - \frac{l(l+1)}{r^2}\right\}P = 0$$

For an attractive field $V(r)$ is negative and $-V(r)$ increases with decreasing r. Solutions are required for negative E satisfying $P(r) = 0(r^{l+1})$ at $r = 0$ and $P(r) \to 0$ as $r \to \infty$. Such solutions exist only for special values of E, the eigenvalues.

More commonly two boundary-value problems and associated eigenvalue problems have been solved by dividing the interval into a finite number of parts and replacing the differential coefficients in the equation by finite-difference equivalents. This reduces the problem to the solution of algebraic equations which are linear if the differential equation is linear. A similar treatment of the eigenvalue problem leads to equations of the form

$$Ax + \lambda Bx = 0$$

where A and B are matrices and x is the vector formed by the values of the independent variables at the dividing points. Because of the way in which they are derived, the matrices arising in the linear-equations system and in the latent-root system have a very high percentage of zero elements. Much of the work on the solution of linear equations and the latent roots of matrices has had its origin in such problems.

An interesting problem of this type arose in the analysis of the possibility that a body of homogeneous fluid may act as a self-exciting generator. Solutions of Maxwell's equations for a sphere of electrically conducting fluid in which there are specified velocities were obtained by expanding the velocity and the fields in spherical harmonics to give a set of linear differential equations with polynomial coefficients. The inclusion of all the radial functions up to degree four led to an eigenvalue problem associated with 12 simultaneous second-order differential equations of considerable complexity. Dividing the interval into 10 parts resulted in a matrix equation for λ of the above form in which A and B were of order 110. It proved possible to reduce this to the problem of finding a latent root of a matrix of order 55. The solution of this final form was complicated by the fact that the resultant matrix was unsymmetric and had many complex roots of which the moduli were much larger than that of the required real root.

21.1.5. Integro-differential Equations. Equations of this type occur frequently in the application of quantum mechanics to the theory of scattering processes. The elastic scattering of nucleons by nuclei gives rise to equations of the form

$$\frac{d^2f}{dr^2} + \left[k^2 - \frac{l(l+1)}{r^2} + V(r)\right]f_l(r) = \int_0^\infty K_l(r,r')f(r')\,dr'$$

where f is the scattered amplitude of the lth partial wave. The expressions for $V(r)$ and $K_l(r,r')$ vary with the nature of the nuclear interaction and also depend on the particular reaction under consideration. The kernel K_l on the right tends to zero with increasing r with sufficient rapidity to make it possible to replace the upper bound of the infinite integral by a convenient finite value. The equations were

solved by replacing both the derivatives and the integral by finite-difference expressions using values at the same points in both, thereby reducing the problem to the solution of linear algebraic equations. A typical example of the equations to be solved is given by inserting

$$V(r) = e^{-\frac{1}{4}r^2}K_1(r,r') = 4[\exp(-\tfrac{1}{2}r^2 - \tfrac{1}{4}r'^2) + \exp(-\tfrac{1}{4}r^2 - \tfrac{1}{2}r'^2)]\, F_{l+\frac{1}{2}}(\tfrac{1}{4}rr')$$

where $F_{l+\frac{1}{2}}(x) = (-1)^l(\pi x/2)^{\frac{1}{2}}\, I_{l+\frac{1}{2}}(x)$ and $I_{l+\frac{1}{2}}$ is the usual Bessel function.

21.1.6. Nonlinear Algebraic Equations. The most common problem in this field is the calculation of the zeros, real and complex, of polynomials. Besides being used in those methods for finding latent roots which depend upon forming the characteristic equation, the problem has arisen a number of times in connection with Heaviside's method for analyzing the response of electrical networks. The solutions of such network problems may be represented in terms of the Heaviside operator p by expressions of the form $A(p)/B(p)$, where A and B are both polynomials. To find the explicit solution, the zeros, usually complex, of the polynomial $B(p)$ are required.

The solution of a set of nonlinear algebraic equations forms the central problem involved in the construction of maps from aerial photographs. A series of photographs are taken of the area to be mapped, successive members covering over-lapping areas. By measuring the two-dimensional coordinates of corresponding points on the two photographs the three-dimensional coordinates of any point on either photograph, referred to a common base, may be calculated.

The equations to be solved are

$$\phi_L r_L = \phi_R r'_R + c$$

and

$$r'_R = M r_R$$

where

$$M = \begin{pmatrix} p_1 & p_2 & p_3 \\ q_1 & q_2 & q_3 \\ r_1 & r_2 & r_3 \end{pmatrix}$$

a rotations matrix. The unknowns are the three components of the vector c, the components of M (of which only three are independent), and ϕ_L and ϕ_R. The elimination of ϕ_L and ϕ_R gives

$$\begin{pmatrix} x_L & x'_R & c_x \\ y_L & y'_R & c_y \\ z_L & z'_R & c_z \end{pmatrix} = 0$$

Five corresponding points in the two photographs determine the unknowns, but more are used to give a strong determination. The equations are solved by iteration, using Newton's method, a good initial guess being possible because the airplane flies roughly straight and horizontally.

21.1.7. Partial Differential Equations. In order to deal satisfactorily with problems in this class a fairly large storage capacity is necessary, so that work in this field was restricted in the early stage of the use of machines. Examples of the application of machines to each of the main types of partial differential equations are the following:

Parabolic Equations. An example solved in England is the problem of determining the temperature distribution at regular intervals in a slab of fish embedded in a refrigerant mixture. The problem is treated as two-dimensional, one in time and one in space, and the equation satisfied by the temperature is

$$\rho\sigma \frac{\partial\theta}{\partial t} = \frac{\partial}{\partial x}\left(K\frac{\partial\theta}{\partial x}\right)$$

where K, ρ, and σ are, respectively, the thermal conductivity, density, and apparent specific heat (taking into account the latent heat to be removed during a change of state). All were given empirically as functions of temperature. If the slab is of

thickness $2X$ and x is measured from the center then the boundary conditions are

$$\theta = \theta_0 \text{ (const)} \qquad \text{for } t = 0,\ -X \le x \le X$$

and

$$-K\frac{\partial\theta_s}{\partial x} = N(\theta_s - \theta_e) \qquad \text{at } x = +X,\ t > 0$$

$$K\frac{\partial\theta_s}{\partial x} = N(\theta_s - \theta_e) \qquad \text{at } x = -X,\ t > 0$$

where θ_s is the surface temperature, θ_e the external temperature, and N the surface heat-transfer coefficient. By a transformation of coordinates the problem was reduced to

$$\frac{\partial H}{\partial t'} = \frac{\partial^2\phi}{\partial x'^2}$$

where ϕ is a function of H.

It was solved by dividing the x range into a number n of sections and replacing the derivatives with respect to x, by a finite-difference approximation. This gave $(n+1)$ simultaneous first-order differential equations which were solved by the Runge-Kutta process.

A second application of machines to the solution of a parabolic partial differential equation was in the boundary-layer theory for an incompressible fluid. The equations to be solved were

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = U\frac{dU}{dx} + \frac{\partial^2 u}{\partial y^2}$$

and

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

with the boundary condition

$$u = 0 \qquad v = 0 \qquad \text{at } y = 0$$
$$u \to U(x) \text{ as } y \to \infty$$

$U(x)$ is the main-stream velocity and the form taken was

$$U = U_0 - \tfrac{1}{8}x$$

The method of solution was to replace the derivative in the x direction by finite-difference expressions and other quantities by averages. From the solution u_1 at x_1 the solution u_2 at x_2 is found from the relation

$$\frac{1}{2}(u_2 + u_1)\frac{u_2 - u_1}{x_2 - x_1} - \int_0^y \frac{u_2 - u_1}{x_2 - x_1}\,dy\left[\frac{1}{2}(u'_2 + u'_1)\right] = \frac{1}{2}\left[\left(U\frac{dU}{dx}\right)_2 + \left(U\frac{dU}{dx}\right)_1\right] + \frac{1}{2}(u''_2 + u''_1)$$

where primes denote differentiation with respect to y. Because of the very sensitive nature of the solution it was not practical on a high-speed computer to assume trial values of u'_2 at $y = 0$ and then carry out the integrations until a value of u'_2 was found which gave the correct behavior at $y = \infty$. On the other hand, the equation for u_2 is nonlinear, so that replacing derivatives of u by finite-difference approximations gives nonlinear algebraic equations. The equations were solved by an iterative process in which a sequence of functions W_n, which were approximations to $(u_2 + u_1)$, were found, the $(n+1)$th approximations W_{n+1} being found from the nth by means of a linear differential equation for W_{n+1} which involved W_n. The equation for W_{n+1} was solved by replacing derivatives of W_{n+1} by finite-difference expressions to give linear equations for the values of W_{n+1} at the points of division in the y direction.

Hyperbolic Partial Differential Equations. The isothermal supersonic flow of a gas past a gravitating-point sink has been studied to calculate the rate of accretion of an isothermal gas by a star moving with supersonic velocity.

The equations to be solved are

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho v r) + \frac{\partial}{\partial z}(\rho u) = 0$$

the equations of motion

$$v\frac{\partial v}{\partial r} + u\frac{\partial v}{\partial z} = -\frac{\mu r}{(r^2+z^2)^{3/2}} - \frac{1}{\rho}\frac{\partial p}{\partial r}$$

$$v\frac{\partial u}{\partial r} + u\frac{\partial u}{\partial z} = -\frac{\mu z}{(r^2+z^2)^{3/2}} - \frac{1}{\rho}\frac{\partial p}{\partial z}$$

and the equation of state of the gas

$$p = A\rho$$

The method of solution was that known as the "method of characteristics" in which the values of u, v, and ρ are obtained at the intersection points of grids of two families of curves, the characteristics.

Elliptic Partial Differential Equations. Electronic computing machines have been used on the computation of prognostic weather maps by solving the equations of motion of the atmosphere. The problem was reduced mathematically to the solution of the equations

$$\nabla^2\frac{\partial h_m}{\partial t} + J(h_m,\beta^2 g l^{-1}\nabla^2 h_m + l) + \frac{1}{3}J(h',\beta^2 g l^{-1}\nabla^2 h') = 0$$

and

$$\nabla^2\frac{\partial h'}{\partial t} + K_1\left(\nabla^2 h_m + \frac{l^2}{\beta^2 g}\right)\frac{\partial h'}{\partial t} + K_1\left(\nabla^2 h_m + \frac{l^2}{\beta^2 g}\right)\beta^2 g l^{-1} J(h_m,h') + J(h_m,\beta^2 g l^{-1}\nabla^2 h') + J(h',\beta^2 g l^{-1}\nabla^2 h_m + l) = 0$$

For each value of t the two equations were solved to give $\partial h_m/\partial t$ and $\partial h'/\partial t$ by a modification of the Liebman process which is appropriate to elliptic-type partial differential equations. h_m and h' were then obtained for the next value of t by the integration of the derivatives with respect to time.

21.1.8. The Use of an Electronic Computer in Testing the Design of a Proton Synchrotron. An interesting use of an electronic computer which is not easily classified under a single mathematical heading was in connection with the design of the new proton synchrotron which is being built at Geneva. The diameter of the track in which the particles are being accelerated is about 600 ft, and in order to economize on the size of magnets producing the focusing field it is intended to reduce track width to a minimum. Very extensive computations have been done to examine the perturbations in the free oscillations of the protons due to minor irregularities in the focusing magnets.

Typical of the types of analysis which were used are the two following. In the first a nearly sinusoidal one-dimensional type of oscillation was assumed given by

$$\frac{d^2y}{d\theta^2} + aQ^2\sin\frac{y}{a} = 0$$

where θ is the angular distance round the track. There were roughly 60 oscillations for each complete circulation, and a perturbation was given at the end of each orbit by applying an increment to Q or to $dy/d\theta$. Since the motion had to be analyzed for a very large number of complete orbits and many types of perturbations it was essential that the initial values of y and $dy/d\theta$ at the beginning of each orbit should be calculated very quickly. In the second calculation the paths were traced through alternate focusing and defocusing sections represented by equations of the form

$$y'' + \tfrac{1}{2}\sin 2y = 0$$
$$y'' - \tfrac{1}{2}\sin 2y = 0$$

Again integration had to be carried out over a large number of complete circulations and for a whole system of random perturbations which were designed to simulate misalignments in the setting up of sections and movements of the foundations.

21.1.9. Crystallographic Calculations. The deduction of molecular structure from observations of X-ray diffraction spectra gives rise to a type of calculation known as a Fourier synthesis. This work has been done on electronic computers and leads to calculation of the type

$$\rho(x,y,z) = \sum_h\sum_k\sum_l F(h,k,l)\cos\left[2\pi\left(h\frac{x}{a} + k\frac{y}{b} + l\frac{z}{c}\right)\right]$$

where $F(h,k,l)$ are coefficients derived from observations. A second problem arising in the same context is the location of the maxima of the function $\rho(x,y,z)$ as defined above. This computation has been carried out by the method of differential Fourier synthesis in which points at which

$$\frac{\partial\rho}{\partial x} = \frac{\partial\rho}{\partial y} = \frac{\partial\rho}{\partial z} = 0$$

are located by a generalization of the Newton-Raphson method.

21.1.10. Analysis of Stochastic Processes. Electronic computers have been used to analyze stochastic processes by performing calculations which are essentially a model of the natural process. An example of this is the analysis of the optimum "on-off" ratios for traffic signals under various conditions in order to minimize the waiting time of vehicles. This has been done by calculating the waiting time for various distributions of traffic, satisfying appropriate statistical requirements, with various "on-off" ratios of the traffic lights.

Problems in diffusion which classically lead to partial differential equations have been successfully attacked by setting up a computational model of the original system, assigning probabilities to the various possible moves of a particle in the system, and then having the computer trace the paths of a large number of particles through the model. The results can be interpreted as a solution of the diffusion problem and therefore as a solution of the corresponding partial differential equation.

Extensive tables of random numbers have been developed for use with this method.

Probability models can be set up for partial differential equations which are not directly related to stochastic processes. Actually probability models have been set up for many computational situations including matrix inversion and the finding of eigenvalues. However, the Monte Carlo technique applied to these situations is usually quite slow.

21.1.11. Data Reduction. This category includes a very large volume of computation, generally fairly elementary in nature.

Engine analysis provides a typical example. A model of an engine is set up and readings of temperatures, pressures, etc., are taken for various combinations of engine parameters. These measurements when properly analyzed give a picture of engine performance. However, this involves considerable computation, best done on a computer. The computing generally involves substituting numbers into formulas. There may be evaluation of trigonometric and other transcendental functions, references to tables, and solution of equations.

21.1.12. Specific Areas of Application. Without any attempt to be exhaustive there are listed here several fields in which computers have found extensive application. The applications included and those omitted are not necessarily a reflection of relative importance but reflect rather the interests and experience of the authors.

Mechanical Engineering. In the design of machinery many procedures can be reduced to relatively routine calculation. Cams, gears, turbine blades, and other geometrical devices can be designed by completely automatic programs, and computers are widely used for this purpose.

Bearing loads have been calculated for a variety of engine types. Torsional vibra-

tions of rotating shafts have been studied. In turbine design and analysis very elaborate heat-balance calculations are performed. Digital computers have been used for the computation of programs for digitally controlled machine tools. The nonlinear differential equations arising in elasticity and plasticity have been studied to some extent by digital means.

Electrical Engineering. Analysis of a-c networks has until very recently been exclusively an analog-computer problem. However, it is now often successfully attacked by digital-computer methods, which have advantages in many instances.

Analyses of transmission-line losses have been carried out. Solutions of electromagnetic-wave equations have been applied to antenna design. The field of radar has stimulated the development of information theory and analysis of time series. Autocorrelations and power spectra are calculated on general-purpose and on specially designed computers. In the field of communication theory it is sometimes desirable to cause a digital computer to simulate the action of an electrical bandpass filter. The nonlinear differential equations arising when nonlinear elements are used in electronic circuits have been analyzed by both analog and digital computers.

Optics. Electronic computing is now in use in just about every major center of optical design. The tracing of rays of light through systems of spherical lenses is a classical computational problem. Until the advent of electronic computers optical designers were generally content to trace only meridional rays, that is, rays in a plane containing the optical axis. They dealt only with low-order aberrations, since skew-ray tracing and computation of high-order aberrations represented a formidable hand-computing task. With electronic computers skew rays take just a little longer to trace than meridional rays and high-order aberrations can be readily computed. The ability to obtain so much more information so easily may lead to a complete change in optical-design philosophy.

Chemical Engineering and Chemistry. In chemical engineering numerous applications of digital computers could be cited. Some of these are:

Evaluation of constants of state for light hydrocarbons
Evaluation of performance of rocket, or other combustion design, by calculating equilibrium gas-product composition
Absorption calculations to determine the number of trays and/or the amount of absorber medium required to produce a given absorption
Computation of batch-distillation curves
Calculation of multicomponent distillation columns with heat balancing
Evaluation of performance factors of fuel-oxidant mixtures
Crystal-structure-factor calculations
Computing infrared spectra for quantitative analysis
Fourier techniques in X-ray organic-structure analysis
Light-scattering calculations for colloidal suspensions
Measurement of optical rotatory dispersion in the steroid field

Aircraft Industry. The aircraft industry was among the first users of computing machines. While some of the applications listed below are included in the general problem classifications given earlier, they are mentioned here as common problems in aircraft:

Flight-path problems for aircraft, including guided missiles
Stability and flutter calculations
Airflow studies to determine coefficients of lift and drag
Analysis of aircraft structures
Simulation of aircraft
Design of aircraft engines, propellers, etc.
Elastic-vibration problems

Physics. Early uses of analog and digital computers were made in several areas of physics. The numerical integration of the path of high-energy particles in earth's gravitational field, carried out by Stormer of Norway, is of significance. This work

was continued in this country with the aid of modern computers by Vallarta and Le Maitre.

The atomic-wave-function calculations were first undertaken on computers by Hartree using the Schrödinger equation. In case of high electron velocities the Dirac equations were used later to take into account the relativistic effects.

Other applications include development of a computer program for the determination of unknown crystal structure from powder photographs, Monte Carlo techniques in nuclear-physics problems, and orbit calculations of high-energy particles in an accelerator.

Social Sciences. Significant applications of quantitative methods and of automatic computers are being made in several areas in the social sciences. Here are listed a few examples:

Dynamic analysis of economic equilibrium. Stock or capital requirement of a particular industry is assumed related to its output, giving rise to a linear differential system. Discontinuities in certain of its derivatives occur if capital stock is not decumulated during a downward trend in a business cycle.

Construction of models of economic behavior and estimation of the parameters involved. This involves maximization of likelihood functions of many variables by means of matrix operations.

Other common areas of applications of computers in social sciences include multiple regression, discriminant function, scaling theory, factor analysis, and time series.

Examples in psychology requiring matrix manipulation are (1) determination of maximum-likelihood estimates of a factorial-analysis structure, and (2) determination of distribution of items in difficulty to yield maximum test-ability correlation.

21.2. DATA PROCESSING*

21.2.1. Introduction. *Data processing,* a general term which could reasonably describe most things done by clerks, engineers, and executives as well as everything done by computers, is used here in its newly acquired, very restricted connotation: the putting of names and numbers through the filing, calculating, deciding, and documenting associated with managing a business enterprise.

Data-processing systems, or management-information systems as they have aptly been called, have three different functions:

1. To perform routine record keeping, computation, and decision making in accordance with programs based on precise statements of company policies and procedures
2. To accumulate, as an important by-product of the routine work, a collection of statistical medians, means, extrema, and other measures of effectiveness
3. To serve as a computational tool in an analysis of the various measures of effectiveness to provide a valid basis on which company policies and procedures may be formulated and modified

Thus computers are used in business both to make routine clerical decisions and to assist in making managerial decisions which require some amount of a not-yet-mechanizable factor called judgment. The aids-to-judgment type of application is discussed in Sec. 21.3; this section is concerned only with mechanizing the routine operations in business.

21.2.2. Businesses and Their Data-processing Problems. Any business basically operates by acquiring goods and services, combining these, and disposing of the product, either goods or services, preferably at a profit. Simple as this seems in principle, the details are quite complicated and differ considerably depending on the character of the individual business. Creators of material goods are the *producers* of such things as food, oil, coal, and other mined products, all dependent to some extent on natural resources; and the *manufacturers* of airplanes, automobiles, steel, chemicals, appliances, machines, electron tubes, tools, office machines, forgings, toys, subassemblies, and whatever, who buy raw material and transform it to finished goods. The service businesses are subdivided into classes such as merchandising (the whole-

* By Charles W. Adams.

saling and retailing of manufactured goods), public utilities (including communications, transportation, and the distribution of electricity, gas, and water), government (including such truly enormous business activities as those of the United States Defense, Treasury, and Post Office departments), finance (such as banks, insurance companies, brokerage firms), and the other professions (education, law, medicine, etc.).

Different these businesses certainly are. But from the point of view of data to be processed, the differences are not so important as are the similarities. Naturally, some of the industries require more data processing than others. Except for governmental requirements, for example, a moderate-sized mine or farm might be run almost without paper work, whereas paper work is the whole foundation of many financial institutions. The crucial differences, nevertheless, are not so much those of the industries but rather those of volumes of data coming in, volumes of data to be filed, types of transactions and number of exceptions, and above all the time scale on which the processing must be done.

Four problems which are present to some degree in every sizable business are those of

Payroll—paying for the services of the employees

Payables—paying for the goods of the suppliers

Receivables—collecting for the products sold

Inventory—maintaining a record of available raw materials, finished goods, or salable services and providing data on the need for replenishment

Of these, payroll and inventory are interesting since each is a large and important problem in most businesses and each has many facets not obvious at first glance. Furthermore, payroll illustrates an accounting function in which the output one day does not appreciably affect the behavior of the system and hence affect the next day's input, whereas inventory management can involve decision making in the computer so that the data processor becomes part of a control system. For these reasons, payroll and inventory control are each discussed below in some detail to serve as illustrations of the type of activity which comprises business data processing.

Payroll. The routine sources of data for payroll are time cards showing total time worked (for salaried employees and regular hourly employees a full work schedule is often assumed, and the time card is replaced by an absence report), and job tickets showing time spent on different jobs (these do not occur in every payroll; they are used for cost accounting and for determining the amount of incentive-type bonuses often paid to production workers). More sporadic are changes in basic file data—arising each time an employee is added to the roles, is given a new job classification, gets a raise, is married, has children, moves, changes his payroll deductions, or quits (not to mention myriad less usual changes which may arise).

The file contains, for each employee, basic data which may include home address, age, sex, seniority status, etc., as well as name, social security number, employee number (if any), number of tax exemptions, job classification, numerous deductions (for union dues, bonds, stocks, hospital insurance, life insurance, pension, credit union, company-store purchases, local taxes, garnishments, etc.), and pay rate. In addition to basic data, which change irregularly, there is a collection of current totals for tax accounting and similar purposes. These include total salary or wages paid for the year to date, income tax withheld year to date, social security tax withheld year to date, and perhaps balances due or paid on other taxes, stocks, bonds, and noncollects (an unwithheld deduction due to be collected later, arising where an employee has committed himself to a deduction, for insurance perhaps, but happens not to earn enough during some pay period to cover the corresponding deductions).

The necessary calculations break down into two major tasks, the calculation of gross pay from the time documents and the calculation of net pay from the given gross. Basically, gross pay equals hours worked times earned rate per hour, but there are several added factors: overtime allowances paid at time and a half, double, or triple time depending on hours worked in a 24-hr period, hours worked per week, holidays, and the like; incentive bonuses based on hours worked, group or individual norms,

and pay rate (or fraction thereof which is subject to incentive); and complications arising because in some cases an employee may work on several different jobs at several different base rates and several different incentive factors within one week.

The primary outputs are the pay checks or, where used, the pay envelopes and the results of "denominating" the payroll—accumulated totals showing how many coins and bills of each denomination are required to fill all the pay envelopes. Journal and ledger data for accounting purposes are obtained but may be kept on tape for further processing and never printed out as is.

Trivial as payroll processing may seem in principle, one must not underestimate the length of the computer program which will be required. The first few actual applications of a large computing system, the Univac, to weekly payrolls in plants employing 5,000 to 20,000 hourly paid persons required computer programs of from 15,000 to 150,000 program steps. This results from the multitudinous details and exceptions represented in the accumulation of different rates, incentives, and deductions built up through the years by industrial engineers, industrial-relations men, unions, charities, insurance companies, and the Federal, state, and local governments.

The situation is typical of many data-processing problems. The principles are simple enough; but requirements set by law, convention, whim, or tradition, and not readily susceptible to standardization, make exception the rule and personalities a hurdle of considerable magnitude in establishing a workable system.

Inventory Control. Inventory control is not one problem but several. Whether one deals in automobiles, hardware, food, space on an airplane or a railroad train, or money (as in a bank), one has an inventory about which one is interested in knowing:

1. The value of the inventory (to calculate profits and taxes)
2. The extent of pilferage or other mysterious disappearance
3. The quantity on hand (to know whether a given order can be filled)
4. When to place an order to replenish the stock of each given material
5. How much to order

In most businesses, the first two functions are handled by taking a periodic physical inventory. Often the business shuts down for two or three days once a year and every item in stock is counted. Valuations and discrepancies are then computed at leisure during the next several months. Questions of whether a demand can be filled and whether stock needs to be replenished are answered by examining the shelf on which the stock is kept. There is in these cases relatively little potentially mechanizable data processing in the inventory-control problem.

Inventory Tally. In some cases it is desirable, and in some cases mandatory, that a running record or tally of the stock on hand be maintained. Purveyors of style goods and seasonal goods, for example, must know from day to day how each stock item is moving in order to restock those in demand while the demand still lasts. And any service industry operating on a by-appointment basis, such as the airlines or the Pullman Company, must keep a tally of its inventory by the very nature of the business. Maintaining a tally to permit answering questions regarding available goods (or space on airplanes or trains), or to provide a basis for determining when and what to restock is commonly called inventory control even though it is not control in its usual present-day sense of influencing the process at hand.

Inventory tallying, whether accompanied by control or not, is then, like payroll, an accounting operation. It differs, however, from payroll in one very important respect—in the allowable time delay between the occurrence of an event and the processing of the information concerning it. In dealing with a weekly payroll, for example, a 2- or 3-day delay is usually permissible after the end of a week before the pay checks must be ready, and there is absolutely no need, in general, to process any data concerning a given week until the end of the week. (There is of course great urgency that the processing all be completed within the allotted two or three days at the close of the week. All payroll personnel live in mortal dread of failing to meet a payroll on time, a calamity which fortunately befalls very few.) Thus payroll lends itself to, in fact cries out for, being processed in batches once a week (or once every two weeks, half month, or month according to the custom of the company).

Inventory tallying for *purely* accounting functions, neither controlling the stock on

hand nor answering questions about it, permits tremendous delays without dire consequences. Most data-processing problems in inventory applications arise because the tallying is being done to permit interrogation or control or both. In these cases the time element becomes quite critical. Determining the availability of space on a given airline flight, for example, must be done rapidly and accurately if the airline is to retain customer good will, and hence its competitive position. Here the delay between the sale of a seat and the recording of it in the inventory tally system must be very small—preferably of the order of a few seconds or less.

Saving up all the data for a week, a day, or even an hour and processing it in a batch is not always feasible since it does not meet the requirements of the problem. Note, however, that where people are involved, as they characteristically are in business problems generally, delays of a few hours are often excessive but delays of a few seconds seldom are. Designers of elevators, telephones, and other such service systems find 15 to 30 sec a maximum tolerable delay. In dealing with certain types of customers or with members of the same organization, circumstances often permit a delay of several hours or even several days without seriously inconveniencing anyone. Since greater speed usually entails greater cost, it is important to determine what delay is permissible and what increased cost a decreased delay is worth.

Inventory Management. Inventory control in the sense of inventory management rather than of inventory tallying mechanizes the determination of when to restock and what amount to restock. The object of any inventory-control system is to provide material when it is needed at the least possible total cost. Since carrying too much inventory costs money (return on investment, cost of insurance, charge for storage space, allowance for possible obsolescence, etc.), while not carrying enough inventory costs money also (increased likelihood of a stock out with attendant expediting expense, delay, loss of sale, loss of future sales, etc.) the theory of inventory management is complicated and not yet perfected.

Determining when and what are questions which, although somewhat interrelated, are usually dealt with separately. Both depend on a prediction of future requirements. Multiplying the estimated stock required each day by the number of days which replenishment normally requires gives a reorder level—except that this makes no allowance for uncertainty in the predicted requirements or in the predicted delivery time of the replenishing supply. A safety stock to allow for these uncertainties is ordinarily added to the reorder level just mentioned. Operations research is just starting to develop useful procedures for predicting requirements or for determining optimum safety stock. There is a widely accepted formula for determining the quantity which should be ordered to replenish stock in some cases. This economic lot size, derived to minimize the total operating cost comprising carrying cost plus reordering costs, is

$$\sqrt{\frac{\text{Quantity used (in units per year)} \times \text{ordering and setup costs (in dollars)}}{\text{Unit cost (in dollars per unit)} \times \text{carrying-cost rate (in dollars per dollar per year)}}}$$

Despite the lack of a complete, sound, practical theory of inventory management, there is great potential dollar value in replacing present hit-or-miss manual systems by even a partially satisfactory system carried out reliably, inexpensively, and expeditiously on automatic data-processing equipment.

Inventory Data Processing. Inventory problems, like payroll and almost all other business problems, involve processing input data against a file, updating the file, performing certain computations, making certain decisions, and producing certain reports. The differences between inventory and payroll are these:

1. The *on-line* nature frequently precludes batching (whereas payroll is inherently a batch-process job).

2. When batching is possible, be the data collected and processed every minute, every hour, or every day, relatively few of the stock accounts are likely to be affected by a batch of inputs (whereas in payroll most employee accounts are).

3. Inventory management is more complex, both theoretically and computationally, while inventory tallying by itself is much simpler than payroll.

4. Whereas payroll is a clearly defined problem differing relatively little from company to company, inventory control is imperfectly defined or understood and its requirements differ widely.

The regular inputs consist of notification of sale or issue of some quantity of a specified item, receipt of some amount of replenishing stock, placement of an order, correction of on-hand balance to compensate for lost material, and inquiries as to the balance on hand or on order. In such cases as airline seats, receipts and orders do not appear explicitly since a scheduled flight generally has a given number of seats which cannot, except in periodic schedule changes, be increased or decreased. When inventory management is involved, the system must deal with such variables as unit cost, setup cost, and lead time (time required between the placing of a replenishing order and the expected delivery of it). As always, there is the question of entering new accounts and deleting old ones.

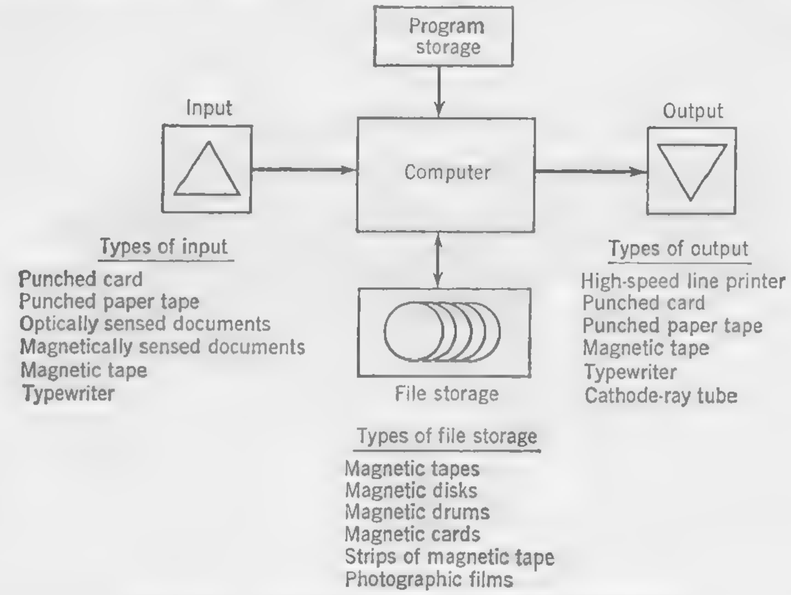


FIG. 21.1. General structure of data processing.

The file, in the case of tallying only, contains only the balance on hand and perhaps the balance on order. When management is involved, the factors just mentioned are required in addition to data about past usage, etc., which are accumulated by the computer primarily for use in the system.

The outputs may be merely the answers to the questions asked, a list of low balances and/or a list of all balances. When management is involved, lists of amounts to be ordered, orders to be expedited, and overdue orders may be among the outputs. Reported less frequently are likely to be routine accounting data such as dollars worth of stock on hand of each type of stock, dollars worth issued in a given period, dollars worth of open orders, and dollars worth of inventory shortages and overages reported.

21.2.3. Characteristics of Business Data Processing. Business data processing deals with, and usually exerts an influence on, some changing situation in a social environment. It is characterized by inputs from and outputs to people, each input relating to some one of many different items stored in a file of constantly changing historic data. Just as in the payroll and inventory applications which have been discussed in some detail, almost all data-processing problems center around the relating of each input to the proper item in the file, performing any indicated computation, and making any indicated decisions. The computations and decisions result either in altering the filed data or in producing some output information or in both. The general procedure is illustrated in Fig. 21.1.

Looked at in another way, business data processing may be said to consist of

1. *Data transcription* (transcribing data for input to the system and output from it)
2. *Filing* (gaining access to filed data and, if necessary, modifying it or adding to it to bring it up to date)
3. *Computing* (performing the arithmetic and making the logical decisions required in processing each item)

There are, of course, situations in which one input item concerns more than one file item, or in which file items are processed when the only "input" is the passage of time without any event having occurred, or in which several input and/or file items are processed to produce a single output. But the bulk of all data processing is concerned with relating each input item to a corresponding file item.

There are two ways in which input data and file data may be matched with one another. Given a random-access file made of magnetic drums, magnetic cores, or magnetic disks, or given a sequential file of magnetic tapes together with enough time, each input (taken in the sequence in which it originated) may be dealt with by calling up the specified file item and proceeding with the processing. This procedure might be called a continuous or on-line process. Alternatively, the data may be batched, sorted into the same sequence in which the file is kept on tape, and processed in sequence. In this latter case, using batch processing, the sequential tape file is more efficiently utilized. There is no unnecessary waiting in going from one item to the next since each item in the file is read (and processed and altered if necessary) only once in the processing of the entire batch. Techniques for sorting input data into sequence preparatory to batch processing are described in Art. 21.2.9.

Batch processing using a magnetic-tape file usually requires file maintenance (adding whole new items, deleting old ones, and changing basic data which do not change regularly or frequently) and file updating (adding new data, deleting old data, and changing accumulative data, where all the data are inherently transitory in nature and subject to regular and/or frequent change) as well as file reference (finding wanted data in a file). Even if the changes necessitated by file maintenance and file updating did not change the length of the different items or alter their sequence, most magnetic-tape systems are so engineered that no data in the middle of a tape can be changed without destroying the remainder of the tape. And most maintenance and updating do change lengths and sequences. Consequently, tape-file maintenance and/or updating is done by reading the original tape and copying it, with any necessary changes, onto a new tape. Since most data processors can read one tape and write another simultaneously (usually while computing as well), this requirement does not appreciably hamper a batched process. Since every item is recopied every time the file is updated, increased emphasis is, however, placed on the reliability of the reading and writing of tape, since the exposure to possible error is here much greates than in a file where inactive items are left totally undisturbed when the active itemr are processed.

In cases where the volume of data necessitates use of a tape file rather than a core, drum, or disk file, but where input data must be processed too rapidly to permit batching, the necessity for recopying the entire file is a major stumbling block. Unless the file is on several reels of tape and only one tape needs to be recopied, or unless only file reference and not file maintenance or updating is wanted, the recopying requirement effectively nullifies any speed increase hoped for in continuous rather than batch processing. Starting from one random position on a tape and moving to another random position involves moving, on the average, one-third of the total tape length. But if recopying is necessary, the entire tape length must be traveled regardless. In fact, if the sequence of the file is not to be reversed by each reference, the new tape must be rewound before it can be recopied in the next reference; in this case, twice the tape length must be moved for a single reference.

The fact that file updating or maintenance often adds or deletes data is a major difficulty in using random-access core, drum, or disk files. Techniques for key-word condensation and for probability-weighted address computation are currently under development. In the meantime, the random-access files are useful principally where

the size and sequence of file items do not change and where efficiently coded key words can be used (for example, in the Teleregister Reservisor system).

21.2.4. Multitape Files Using Alternate Tape Units. The basic tape data-processing procedure starts with an input tape and a file tape and produces an updated file tape and an output tape. Often the file and sometimes the input and/or output contain more data than will fit on a single physical tape. In this event, more than one tape must be used. Rather than to let the computer stand idle while each tape, once read, is rewinding preparatory to being dismounted and replaced by the next tape, a system of alternating tape units (frequently called a tape swap) is employed. In this the program is simply so arranged that, when the first tape, mounted on one tape unit, has been completely read (or written), that tape is started rewinding and data are taken from a second, or alternate, tape unit on which has been mounted the second tape of the file. By the time the second tape has been processed, the first tape will, in general, have been rewound, dismounted, and replaced by the third tape. Hence the program has merely to alternate back and forth between two tape units in order to read (or write) a multitape file without appreciable delay in changing tapes (see Fig. 21.2).

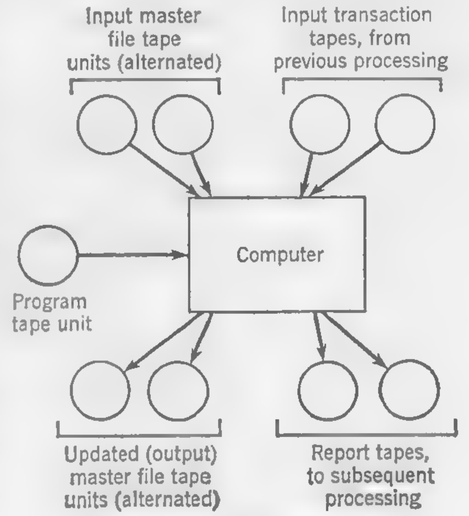

Input master file tape units (alternated)
Input transaction tapes, from previous processing
Computer
Program tape unit
Updated (output) master file tape units (alternated)
Report tapes, to subsequent processing

FIG. 21.2. Run structure.

21.2.5. Unit Records, Fields, Keys, and Trailers. The information about each individual employee in a payroll employee file, about each different stocked material in an inventory file, or about each individual account in any data file is called either an *item* or, more descriptively, a *unit record.* By the same token, the details of any one transaction, event, or change being entered as input likewise constitute an item or a unit record, as does any one related set of output data. While there is no concise definition or universal usage, there is a tendency to use the word *item* in describing inputs or outputs and *unit record* in referring to basic file data.

A punched card is a unit record which is limited to a maximum of 80 (or 90 or whatever, depending on the manufacturer) digits or characters of information, one in each column of the card. The terms *digit* and *character* serve as an informal unit of measure of information. A digit is one of the 10 decimal digits, or alternatively one of the 16 symbols, including the decimal digits, which can be represented by combinations of 4 binary digits or bits. A character is one of the 50 or so symbols on an uppercase typewriter keyboard, including the digits, the letters of the alphabet, and certain print-control functions (tabulation, carriage return, etc.). Thus a digit is represented by 4 bits, as on a four-channel tape; a character is represented by 6 bits, to which is usually added a seventh bit for checking purposes, resulting in a seven-channel paper or magnetic tape.

By analogy with long-standing punched-card nomenclature, a unit record is made up of fields of data. Each field contains one quantity, one identifying code number, or one name or other word description, abbreviated or not. Generally speaking, the length of a field in digits or characters is chosen so as to allow for the longest number or name expected to occur. In the standard, fixed-word-length computers, the choice of field length is influenced, sometimes quite strongly, by the built-in word length. On the other hand, the tape systems associated with some of the individually-addressable-character (less properly called variable-word-length) computers give a free choice of field length, while the tapes in other such computers give a truly variable field length, in which the field length adjusts itself to the actual length of the individual numbers or names. In either case, the internal storage system behaves as a free field length and

not a variable-field-length device. Figure 21.3 illustrates three unit records for a hypothetical payroll employee file, laid out for a fixed-word-length system, for a free-field-length system, and for a variable-field-length system.

A tape file ordinarily consists of many unit records arranged in sequence according to some identifying name or number. The identifying name or number is called the key and occupies one of the fields in the unit record. There is ordinarily no reason for assigning the first field to the key word, but the key normally comes near if not at the start of the unit record. Not uncommonly it is efficient to maintain two or more files of unit records describing different aspects of the same set of accounts, and often the two files are arranged in different sequences, with one field being the key in one set of unit records and another field the key in the other. This situation is especially common in sequences of output data, where the same data are to be printed and/or summarized in two or more different sequences. Hence, given a unit record, one cannot

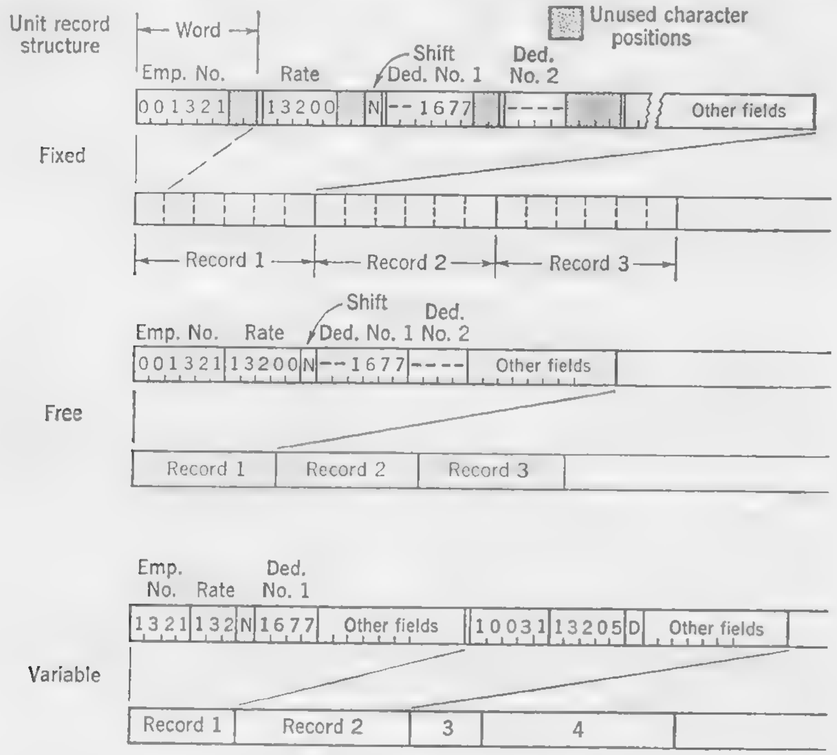


FIG. 21.3. Typical unit records.

a priori specify which field contains the key, for this depends on the way in which the record is to be processed and not on the record itself.

When a unit record contains more than 80 characters, it cannot be recorded on one 80-column card. Standard punched-card practice in this case is to employ a *trailer* card, which is a second card intended to follow the first and identified by the same key. Tape systems and computers seldom encounter this situation since the maximum length of unit record which can be handled at one time usually ranges from several hundred to several thousand characters, depending on the system at hand. Often, however, a situation arises in which a unit record should contain a list of indefinite length. For example, in controlling inventory it is often convenient to know, for each item stocked, the quantity specified in each unfilled order and the date on which each order is due. Some accounts may have many open orders to list while other accounts have none. It would be extremely wasteful to allow, in each unit record, two fields for each of some maximum number of orders. Instead, the open orders are often carried as two-field trailer items which are carried along behind the main unit record

(or *header*) item. Since the number of trailers in a given account record varies, this procedure is a form of variable record length achieved through programming. The free- or variable-field-length tape systems usually have built-in facilities for handling variable record lengths as well, but just as in the case of variable field length on tape, the internal storage is less flexible and space must be allotted for the largest possible

I. The nature of the processing
  A. The extent and frequency by which input data alter the filed data in each file
    1. Percentage of file records which are active (that is, have a corresponding input item) in each processing period
    2. Need for updating the records
    3. Frequency with which changes make file maintenance necessary
    4. Number of file records changed during maintenance
  B. The complexity of the basic processing of an input item with a file record
    1. Arithmetical
    2. Logical
  C. The exceptions
    1. Number of different types of exceptions
    2. Frequency with which each type of exception occurs
    3. Complexity of each type of exception
  D. The amount of processing required in preparing the inputs
    1. Editing within an item
    2. Sorting in sequence (or checking the sequence)
    3. Collating the data
      a. To combine inputs from several sources
      b. To select only the pertinent inputs
  E. The amount of processing required in preparing the outputs
    1. Editing within an item
      a. Rearranging
      b. Zero suppression
    2. Rearranging into different sequences
    3. Summarizing
II. The volume and structure of the data
  A. File data
    1. Number of digits or characters per unit record
    2. Number of unit records per file
    3. Number and length of any trailers
    4. Number of different files
  B. Input data
    1. Number of items per unit time (usually per processing time)
    2. Effective average length of the items
  C. Output data
    1. Number of different reports
    2. Frequency of each report
    3. Number of printed lines for each report
III. Speed and other special requirements
  A. The delay between the arrival of data and the complete processing of it (minutes or days)
  B. The arrival pattern of the data (random or periodic)
  C. The desirability of having the source documents (from which input data are derived) be physically part of the output (as in processing bank checks)

FIG. 21.4. Factors in determining the magnitude of a problem.

number of trailers. When the trailers are dealt with one at a time by the program, internal storage capacity for only one trailer can be made to suffice.

21.2.6. Factors Determining the Magnitude of a Problem. In the outline of Fig. 21.4 are indicated the various major factors which enter into the determination of the magnitude of a problem. The three major headings, if interpreted broadly, suffice to define completely the magnitude of any data-processing problem. In any given problem, one of these three considerations will be dominant. The detailed factors comprising each major consideration are so numerous, interrelated, and diffi-

cult to define or measure, however, that the outline merely suggests the highlights and does not show relative importance or interrelationships of these details.

In a routine problem in which the special requirements do not preclude use of magnetic-tape files, it is the total number of digits or characters in the file that usually is most important in determining total time requirements. When the processing is relatively complex, when most file records are processed in each run, and when the tape is relatively fast and/or the computing device relatively slow, the speed of the main run may be "computer-limited" rather than "tape-limited," as is more often the case. In the tape-limited case, over-all processing time is usually less than twice the time required to read and copy the main file. This time equals the digits per unit record times the number of unit records in the file, divided by tape reading-writing speed in digits per second. When the process is computer-limited, the time is estimated by multiplying the average time to process a unit record by the number of unit records on file. Determining this average time requires that the processing be at least roughly charted and coded, either on paper or in the mind of a skilled programmer.

In some instances the processing or the exceptions are so complex that the processing program cannot all be stored at one time in the program-storage element, and the one main run becomes two. (In the tape-limited case, making two passes in place of one doubles the total time.) In other cases, the volume of input and/or output data, or the complexity of editing the input or output is great enough to contribute heavily to the total processing time, but this is not usual.

21.2.7. Criteria for Evaluating Equipment.[1] An evaluation of automatic data-processing equipment requires the establishment of criteria or measures of effectiveness on which to base the comparison. What each potential user wants is to get his job done at the least cost. The best general basis for comparison would then seem to be the cost for performing some typical job or jobs on each of the available equipments.

Unfortunately, there is no typical job. What is worse, there is not even a clear-cut job unique to a given user. Different systems will accomplish what is wanted in different ways, giving different intangible, hard-to-evaluate extra features. What value, in dollars and cents, does one put on speed, reliability, an extra piece of summary information, an easy-to-read report? The problem is not unique. Similar unmeasurable factors of course occur in selecting a new automobile, a new factory site, a new systems man, or any other commodity in a competitive market.

What one would like in all these cases is a set of numbers describing one's needs, a set of numbers describing equipment from which one can choose, and a formula into which the two sets of numbers can be substituted to give a true measure of effectiveness. No such numbers and no such formula are known to the authors. Figure 21.5 does, however, list most of the criteria which would probably be important independent factors in such a formula if one existed. Each of these factors refers to the system as a whole, and each is therefore made up of numerous contributing factors, many of which are noted in the outline.

21.2.8. The Need for Sorting. Sorting data into some predetermined sequence is one of the principal functions in most manual and punched-card data-processing systems. The primary reason for sorting is of course as a preliminary for filing in batches. But there are other reasons as well.

Summary distributional information is customarily produced by sorting into the categories for which totals are wanted, summarizing, and sorting back into sequence again. For example, a company selling 200 different lines of products collects sales data in sequence by salesman but wishes also to know total sales for each product line. Cards describing each sale are sorted in sequence by product line; the 200 totals are obtained one at a time; and the cards are re-sorted into sequence for refiling, or for another distribution (by customer, for instance).

Sorting has even been used as a preferable substitute for calculating rates. For example, rather than compute charges for each meter reading, one utility sorts all readings by quantity and applies a precomputed charge to each group of identical readings. The charge tickets must then be re-sorted into whatever alphabetical or geographical sequence the ledgers are maintained in.

The reason for the preoccupation with sorting thus appears to be twofold. First, sorting is often one way and sometimes the only way to achieve the desired results. Second, manual sorting and especially punched-card sorting can be done by well-established techniques quickly, easily, and inexpensively.

Electronic data-processing equipment can be built or programmed to sort large volumes of data recorded on reels of magnetic tape. It can do the sorting faster than it can be done by punched-card techniques. But it is often considerably more expensive to sort using magnetic tapes than to sort using punched cards.

I. Cost
  A. Central computer—rental, or purchase and maintenance
  B. Tape units and their controls—same
  C. Input and output units and controls—same
  D. Air conditioning
  E. Site preparation, furnishings, physical installation, etc.
  F. Tapes (100 to 1,000 at $30 to $80 a reel), cards, etc.
  G. Preparation and improvement of programs
II. Speed
  A. Internal arithmetical and logical operations, including
    1. Effect of index registers
    2. Effect of latency reduction by buffers, etc.
  B. Tape speed, including the effect of
    1. Multiplexed operation—independent reading, writing, rewinding, searching, merging, sorting
    2. Fast rewind and/or search
    3. Ease of manual mounting, dismounting, storage of tape
  C. Input and output devices
III. Capacity
  A. Number of decimal digits of primary storage
    1. Length of instructions
    2. Efficiency of handling alphabetic information
  B. Number of decimal digits in auxiliary (drum) storage, if any
  C. Number of tape units manageable (NOTE: The number of digits per tape affects the number of physical tapes and of manual tape changes required; hence it affects the cost and speed but not the capacity)
IV. Reliability
  A. Freedom from undetected errors, caused by
    1. Computer
    2. Tapes
    3. Input-output
    4. Operator
  B. Efficiency, per cent of total available time required for
    1. Scheduled preventive maintenance
    2. Unscheduled corrective maintenance
    3. Programmed checking and rerun procedures
V. Ease of programming, of operating
  A. Time and talent required for planning, for coding, for operating
  B. Likelihood of mistakes in coding, in operating
  C. Ease of locating and correcting mistakes ("debugging")
VI. Flexibility
  A. Relative efficiency in different types of problems
  B. Ease of switching from program to program
VII. Physical size and appearance
VIII. Availability
  A. Date of first installation of this type of equipment
  B. Date of delivery promised to user in question
IX. Manufacturer's reputation
  A. Dependability and speed of maintenance and repair
  B. Publicity and prestige attendant on the equipment
  C. Sales appeal

FIG. 21.5. Primary criteria for digital-equipment evaluation. (*From an article by C. W. Adams in Special Report No. 3, "Electronic Data Processing in Industry," copyright by the American Management Association, Inc.*)

The difficulties of tape sorting can sometimes be overcome by avoiding the process entirely. Utility charges can be computed rather than looked up. Totals by product line can often be accumulated by reading the data sequentially only once and adding each amount into its proper accumulated total, making use of the thousand or more random-access-storage locations available in most electronic data-processing machines. Frequently, sorting is used to rearrange data for some special report which could equally well or perhaps better be prepared in some more easily produced sequence if not eliminated entirely. Finally, source data and/or final reports which happen to be prepared on punched cards for other reasons can be sorted on punched-card sorters.

Using punched-card sorters in preference to tape sorters is not always the best procedure. It is almost never economical to sort a tape file by punching cards from it, sorting them, and then reconverting the cards to tape. Even with source data which are originated on punched cards, punched-card sorting will not be the most economical unless the nature of the job makes the card sorting much cheaper on the face of it and unless the sorter and operator are readily available. For one thing, good practice requires that, even after data have been sorted on card sorters prior to being recorded on tape, the sequence on tape be checked by the computer to detect any operator's mistakes which might have occurred during the sorting or the conversion of cards to tape. The choice of method of sorting in preparing final reports or other outputs requires, just as in planning the sorting of input data, that each case be judged on its own merits.

The actual cost of sorting data which have been magnetically recorded on tape depends on:

1. The size and number of items to be sorted
2. The size and location of the key
3. The speed and number of the available tape drives
4. The buffering, internal storage capacity, and operating speed of the computer
5. The technique used in sorting

The choice of technique in turn is dependent on the characteristics of the items, keys, tapes, and computer.

21.2.9. Sorting Techniques. Because of the basic importance of sorting to data processing, a rather long and detailed, though necessarily very incomplete, description of sorting techniques is presented in this section. The results are summarized in Art. 21.2.10.

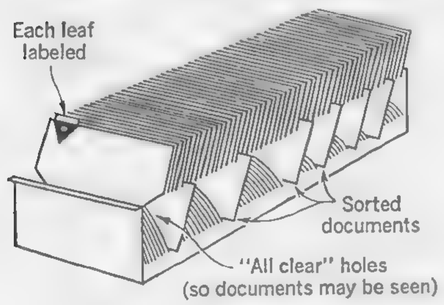


FIG. 21.6. Leaf sorter.

There are three fundamentally different approaches to sorting data into sequence. These are usually described, respectively, as *digital* sorting, *merge* sorting, and *internal* sorting. Internal sorting, as used here, is a catch-all term to describe every orderly procedure applicable only to random-access storage and not to tape storage. Merge sorting is a process which is especially useful in tape sorting and not commonly used except for sorting on tapes. Digital sorting, which is more descriptively called pigeonhole sorting, is sometimes used in sorting on tapes, usually used in manual sorting, and always used in standard punched-card sorting.

An understandable explanation of any sorting technique is surprisingly difficult to make without using lengthy examples, preferably in the form of a step-by-step demonstration. The reader who wishes to familiarize himself with the sorting techniques about to be described is therefore urged to peruse the examples given and then to experiment further for himself using a deck of playing cards.

Digital Sorting. The most obvious way to sort a group of documents is first to set up for each different key a pigeonhole, a space on a table, a space between two leaves in a leaf sorter (see Fig. 21.6), or the equivalent. Then the keys on each successive document may be examined, the document put in the corresponding pigeonhole, and the process thus completed in a single pass. This works superbly if (1) the number of possible keys (the range of the key) is small and if (2) each pigeonhole is big enough to hold all the documents it might receive if more than one document has the same key.

The distribution of sales by product line mentioned earlier is an example in which neither physical documents nor corresponding sets of information are sorted but which nonetheless amounts to a single-pass pigeonhole sort in which only total sales per product line are accumulated in the 200 individual product-line pigeonholes. Two hundred is a very small range of key. Large magnetic-drum storage systems such as the American Airlines Reservisor or the Univac File computer "avoid" sorting by providing enough pigeonholes for certain moderately large ranges of key and by providing some method of key-word condensation. But, whether one is sorting documents mechanically or items electronically, many sorting problems will involve a greater range of the key than can economically be provided for in the random-access storage systems now available.

The solution to handling larger ranges of key is to perform the sort in several steps—several passes through the documents or items. A person anxious to sort a deck of playing cards, for example, is likely to start by first sorting the deck into four parts by suit. Then each pile, containing a single suit, is sorted by distributing the individual cards into their proper slot by face value, then picking them up in sequence. Sorting first into major groups to obtain more readily handled groups is called block sorting. It is useful in manual sorting and in distributing a large sorting job among several electromechanical sorters preparatory to machine sorting, but it is just the reverse of a true digital sort.

In a true digital sort, the first distribution into several pigeonholes is done on the basis of the smallest digit of the key. In sorting a deck of playing cards, one might distribute the deck into 13 piles, with all the twos in one pile, threes in the next, and so on. Then the entire deck would be collected back together with the twos on the top, threes next, etc. Now if the deck is distributed into four piles by suit, each suit will be in sequence since the twos of every suit came before any of the threes, and so on. Elementary as this is, the simplest way to convince oneself is to try it. The advantage to starting with the least significant "digit" first lies in the fact that all the cards can be reassembled and the second pass made on the complete deck, not on individual parts of it as was the case when the cards were distributed first into major groups by the most significant digit. Ordinarily, sorting numerical keys is done by sorting first on the smallest decimal digit, then on the next smallest, and so on.

As a demonstration of a simple digital sort and (at the same time) an illustration of the possibility of using a nonobvious digital base for the sorting, consider how a deck of playing cards might be sorted digitally by making two passes, distributing the cards into eight different pigeonholes each time (rather than 13 in the first pass and 4 in the second as was described above, and rather than 10 each time as is usual in numerical sorting using electromechanical sorters). Notice that two passes, each into eight categories, should suffice since splitting the deck eight ways on the first pass and eight again on the second gives 64 possible outcomes for a given card while there are only 52 in a deck. By the same token, the sorting could not be accomplished in two passes using only seven pigeonholes since $7 \times 7 = 49$, which is less than 52. For the 8-way digital sort, the pigeonholes can be labeled 8's, 9's, 2's and 10's, 3's and jacks, etc., as shown in Fig. 21.7. There are several other labelings which would work as well. These particular assignments were chosen only because they correspond to naming the pigeonholes 0, 1, 2, 3, 4, 5, 6, 7, respectively, and then allocating cards according to the remainder after dividing by 8, counting jacks as 11, etc. (so that, for example, queens, equaling 12, leave a remainder of 4 and go with the fours). If the aces are to sort out low instead of high, they should be included in the "one" pile with the nines instead of in the "six" pile. On the second pass, the pigeonholes are labeled "spade 8 to ace," "spade 2 to 7," etc. (or "8 to king" and "ace to 7" if the aces are to sort out low).

A 52-card deck can be sorted in three passes using only four pigeonholes (since $4 \times 4 \times 4 = 64$, which is greater than 52, the range of the key). The first pass would put the 4, 8, Q in the first slot, the 5, 9, K in the next, the 2, 6, 10, A in the next, and

Sequence of the deck to be sorted before first pass:

S3	CQ	SJ	SQ	SA	S6	S7	H4	H7	SK	H5	D9	S8
C8	CK	D6	D3	HQ	DT	C2	H6	H9	C9	CJ	HT	HJ
C6	CT	S9	ST	D5	D8	D2	DA	S2	S5	H2	C3	C4
C5	CA	HA	H8	H3	DJ	DQ	C7	D7	D4	DK	S4	HK

After first pass: Cards taken in sequence above (left to right, top row first) and placed in proper pigeonholes below:

Pigeonhole Number	Label	Cards in Sequence							
0	8's	S8	C8	D8	H8				
1	9's	D9	H9	C9	S9				
2	2 & 10's	DT	C2	HT	CT	ST	D2	S2	H2
3	3 & J's	S3	SJ	D3	CJ	HJ	C3	H3	DJ
4	4 & Q's	CQ	SQ	H4	HQ	C4	DQ	D4	S4
5	5 & K's	SK	H5	CK	D5	S5	C5	DK	HK
6	6 & A's	SA	S6	D6	H6	C6	DA	CA	HA
7	7's	S7	H7	C7	D7				

After second pass: Cards taken in sequence (left to right, top row first) from pigeonholes above and placed in proper pigeonholes below:

Pigeonhole Number	Label	Cards in Sequence						
0	Spade 8-A	S8	S9	ST	SJ	SQ	SK	SA
1	Spade 2-7	S2	S3	S4	S5	S6	S7	
2	Heart 8-A	H8	H9	HT	HJ	HQ	HK	HA
3	Heart 2-7	H2	H3	H4	H5	H6	H7	
4	Diamond 8-A	D8	D9	DT	DJ	DQ	DK	DA
5	Diamond 2-7	D2	D3	D4	D5	D6	D7	
6	Club 8-A	C8	C9	CT	CJ	CQ	CK	CA
7	Club 2-7	C2	C3	C4	C5	C6	C7	

Cards are now in desired sequence, left to right, bottom row first.

FIG. 21.7. Digital sorting (base 8).

the 3, 7, J in the last. The second pass would put the 2, 3 in the first, the 4, 5, 6, 7 in the next, the 8, 9, 10, J in the next, and the Q, K, A in the next. The third pass would distribute by suit.

Applied to manual sorting of playing cards, the true digital sort seems absurdly artificial and cumbersome. Teamed up properly with any of several kinds of punched cards and the proper mechanical or electromechanical gadget, however, it is an exceedingly powerful technique—so powerful that most electronic computers suffer by comparison. The two principal mechanical techniques are:

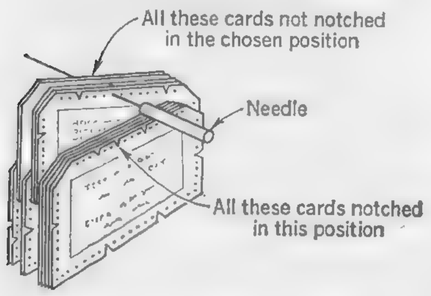


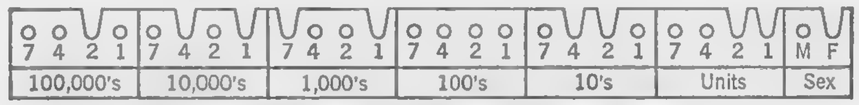
FIG. 21.8. Needle sorting.

1. Needle sorting of edge-punched cards
2. Electromechanical sorting or mechanical sorting of conventional punched cards (e.g., IBM cards)

Both of these are so elegant in their simplicity and effectiveness that, though they may seem out of context in a manual of electronic-computer techniques, they should be understood by everyone involved in any extensive automatic data-processing activities.

Needle Sorting of Edge-punched Cards. Needle sorting is a way of separating properly prepunched cards into two groups. The comment above about sorting playing cards in three passes using only four pigeonholes could be generalized to

sorting the same cards in six passes using only two pigeonholes. For strictly manual sorting this would be ridiculous, but with edge-punched cards and a needle (see Fig. 21.8), it is very effective indeed. Edge-punched cards are manufactured with a series of small round holes along one or more edges. The key or keys by which the card is to be sorted are punched into the card by notching out appropriate ones of the holes along the edge (see Fig. 21.9). All cards to be sorted are stacked together and a knitting needle is passed through the hole in the "ones" position of the least significant of the key. When the needle is lifted, all the cards with notches in that position drop off and are put in the rear of the deck. The needle is put through the "twos" position, lifted, and the cards which drop are put in the rear, without disturbing their sequence relative to one another. The process repeats for the "fours" and then for the "sevens" position of the least significant digit, then on into the next digit, and so on. The power of the methods lies in the speed with which the notched cards can be



Notches represent the number 258,063 and word "female"

FIG. 21.9. Edge-punched card.

separated from the unnotched ones in each column. For certain types of files which change infrequently and are frequently searched for different categories or for certain specific items—that is, files such as library references or personnel files which are referred to by different keys at different times—the edge-punched card is, for all practical purposes, the fastest and by all odds the cheapest method known.

Electromechanical Sorting of Conventional Punched Cards. Much more frequently used than the edge-punched card is the conventional punched card (see Fig. 21.10). The IBM card is divided arbitrarily into 80 columns, any or all of which may be punched in one (or two or sometimes three) of twelve different row positions. The rows are numbered, from top to bottom, 12, 11, 0, 1, 2, 3, . . . , 9. The 12 and 11 holes serve to identify special functions, negative numbers, and alphabetic symbols and should temporarily be neglected. One of the simplest yet most powerful devices for processing punched cards is the sorter (see Fig. 21.10). A sorter examines any single column in each card successively and deposits the card in one of 13 pockets depending on whether the first hole encountered is in the 12 row, the 11 row, . . . , the 9 row, or whether no hole is present at all (a reject). There is provision for suppressing any rows desired from the examination, and cards not punched in any of the nonsuppressed rows end up in the "reject" pocket.

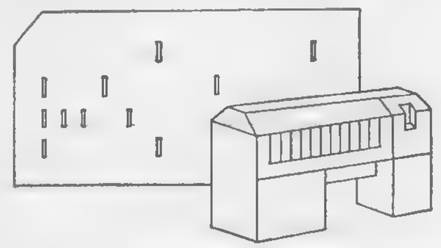


FIG. 21.10. Electrical sorting machine and punched card.

A card sorter in operation gives the impression of deciding which pocket to deposit each card in as the cards move along rollers above the pockets. Not apparent at first glance are a pack of 12 flat metal leaves called chute blades, one terminating above each pocket. The decision as to which pocket each card belongs in is made not above each pocket but rather when the card first leaves the feed hopper and passes under a single brush (see Fig. 21.11). The brush has been manually positioned to the desired one of the 80 column positions. When a hole in the card allows the brush to touch the conducting roller, an electromagnet is energized which lowers an armature and allows all the chute blades not yet touching the card to drop. The card then rides up onto the proper chute blade and is carried along the top of the sorter until it is inevitably tumbled into the proper pocket.

Sorting cards according to a numerical key is done by first positioning the brush to the column containing the least significant digit of the key. After the cards have gone

through the sorter, they are manually collected from the various pockets, zeros below ones below twos, etc., and put back in the feed hopper. The brush is moved to the next more significant column and the sorting repeated. Sorting by a six-digit key requires six passes. The cards move through the sorter at speeds of 450 to 1,000 cards per minute (7.5 to 16.7 cards per second) depending on the price of the sorters, which cost in the vicinity of $1,000. On the highest-speed sorters, 10,000 cards can be sorted on a 6-digit key in 1 hr. Under the worst conditions, the same sorting on a Univac I computer might take 15 min and cost more than ten times as much for the job.

Alphabetic data on punched cards require special treatment dependent on the coding of the cards. Briefly, with IBM cards, the cards are sorted on a given column with the 12 and 11 rows ignored. The zeros are set aside; the rest are collected and run

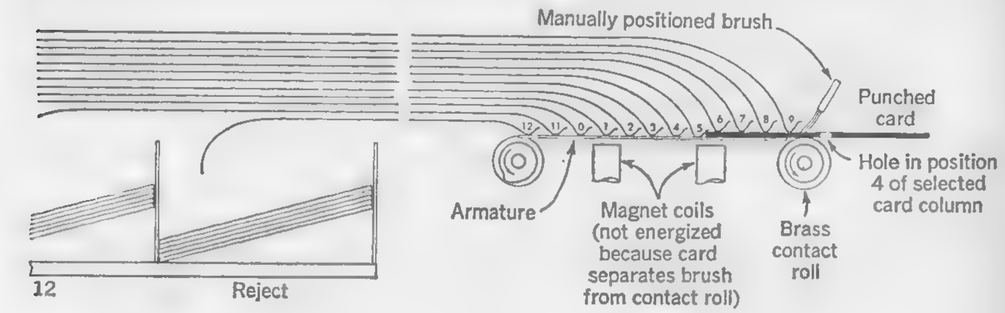


Fig. 21.11. Operating principle of punched-card sorter.

again with every row except that the 12, 11, and 0 rows are ignored. The numerical cards end up in the reject pocket, letters A through I in pocket 12, J through R in pocket 11, and S through Z in pocket 0, each group in sequence (from the first pass).

Digital Sorting Using Magnetic Tapes. Digital sorting using magnetic tapes is performed in much the same way as punched-card sorting. Each item of information is copied from the input tape (analogous to the feed hopper) onto the proper output tape (corresponding to one of the pockets in a sorter) rather than being physically moved, with the information still intact, from input to selected output as in a punched-card sorter. Aside from the fact that information is copied rather than transported from input to output, there are three more trivial differences due to certain peculiarities or limitations of tape units. These are set forth briefly in the following three paragraphs.

Since tape units usually are not efficient in start-stop operation, and since the rate of arrival at each of the outputs (corresponding to the sorter pockets) is unpredictable and uneven, arrangements are usually made to accumulate enough separate information to be sent to a given output to make it worthwhile to start the tape unit and disgorge the accumulated items onto it. This block handling of tape data is conventional in all aspects of tape use.

After one pass in which information has been dispersed onto the several outputs,

it would be wasteful to read back all the data merely to collect it onto one tape for the second pass. Therefore, each of the original outputs is usually treated in its turn as an input, with information being dispersed onto a different set of new output tapes in accordance with the next larger digit of the key. Thus the tapes which serve as output "pockets" in the first pass are usually used sequentially as inputs in the second pass, then serve as outputs again on the third pass, etc. Clearly a base-10 digital sort, requiring 10 pockets, requires 20 tapes if this procedure of alternating inputs and outputs is employed.

Most tape-processing systems do not, for reasons of economy, include 20 tape units. Consequently, when digital sorting is done using tapes, it is usually done to the base 2 or 3 or possibly 4, requiring 4 or 6 or 8 tape units simultaneously available. Examples of nondecimal digital sorts were given in the examples of sorting playing cards.

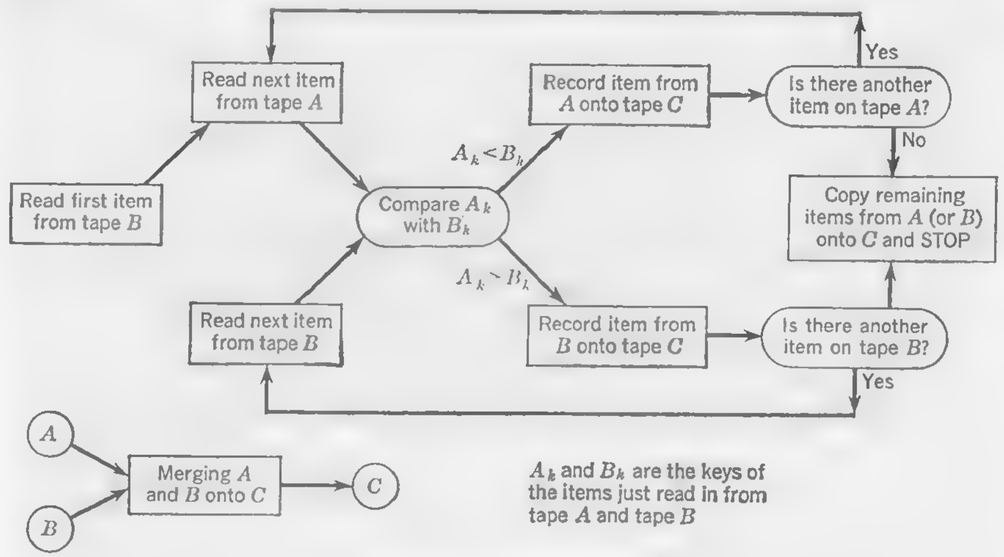

Read next item from tape A
Record item from A onto tape C
Is there another item on tape A?
Yes
No
Read first item from tape B
Compare A_k with B_k
$A_k < B_k$
$A_k > B_k$
Copy remaining items from A (or B) onto C and STOP
Read next item from tape B
Record item from B onto tape C
Is there another item on tape B?
Yes
A
B
Merging A and B onto C
C
A_k and B_k are the keys of the items just read in from tape A and tape B

Fig. 21.12. Merge program.

The number of passes p required when each pass disperses into b tapes or pockets (a base-b sort) and when the largest key number is r, is

$$p > \log_b r = \log r / \log b \qquad \text{(in common, base-10 logarithms)}$$

For example, a 6-digit key gives $r < 10^6$. A decimal ($b = 10$) sort requires $p > \log 10^6/\log 10 = 6$ passes while a ternary ($b = 3$) sort requires $p > \log 10^6/\log 3 = 6/0.4771 = 12.55$, or 13 passes.

Merging. Merging, not to be confused with merge sorting, which will be described in the next paragraphs, consists of combining two or more groups of data, each in sequence by itself, into a single group in the same sequence. Thus the two strings of numbers (*A*) 7 13 14 19 21 24 25 29 43 and (*B*) 4 17 18 25 27 28 31 37 41 when merged yield (*A* merged with *B*) 4 7 13 14 17 18 19 21 24 25 25 27 28 29 31 37 41 43. The two-way merge process is quite simple to mechanize in a computer. If two tapes *A* and *B* are to be merged, the program (shown in Fig. 21.12) involves comparing the keys of the items from each of the two inputs, selecting the smaller, putting it on the output, replacing it by the next item from the input whence the selected item came, and repeating the process until both inputs are exhausted.

Merging 3, 4, 5 (or more) tapes (a three-way, four-way, or five-way merge) is done in precisely the same way except that selecting the smallest requires an increasingly large number of two-way comparisons.

Merge Sorting (Planned String Length). Merging two groups, or strings, each, for example, consisting of a 100 items arranged in sequence by some key, yields a single

sequenced string of 200 items. This fact that a two-way merge effectively doubles the number of items in a single sequence means that merging is itself an ordering procedure, one which can be used to arrange information in sequence. The relative merits of merge sorting and digital sorting are considered in the next section. The mechanization of merge sorting is best understood by example.

Given a well-shuffled deck of playing cards (to be put into the sequence SA SK SQ .... S2 HA .... C2 as in the digital-sorting examples), one can proceed by first halving the deck. The halves might look as in Fig. 21.13(*a*). These two sets of cards cannot be merged in the normal sense since each set is not in sequence to begin with. However, each half can be considered as 26 separate "strings," each consisting of one card. These 26 strings on each tape can be merged separately in descending sequence into 26 strings of 2 cards each, with the resulting strings being recorded alternately on each of two tapes. Thus, in Fig. 21.13(*a*) the first card from

S4 CQ H3 S3 CK H5 SQ D2 D9 H2 DT S7 SJ C7 C5 S2 H7 D6 C2 C4 CJ D4 DA S5 HQ SA
ST DJ H9 S6 C6 HJ D5 HK DQ HT H8 SK D3 CT HA C9 C3 S9 CA C8 H6 D8 DK S8 D7 H4

(*a*) Initial arrangement divided into halves—strings of $1 = 2^0$.

ST S4, H9 H3, CK C6, SQ D5, DQ D9, H8 DT, SJ D3, HA C5, H7 C3, CA C2, H6 CJ, DA DK, HQ D7
DJ CQ, S6 S3, HJ H5, HK D2, HT H2, SK S7, CT C7, S2 C9, S9 D6, C8 C4, D8 D4, S8 S5, SA H4

(*b*) After first pass, in sequence by pairs—strings of $2 = 2^1$.

ST S4 DJ CQ, HJ H5 CK C6, HT H2 DQ D9, SJ D3 CT C7, S9 H7 D6 C3, H6 D8 D4 CJ, SA HQ H4 D7
S6 S3 H9 H3, SQ HK D5 D2, SK S7 H8 DT, S2 HA C9 C5, CA C8 C4 C2, S8 S5 DA DK,

(*c*) After second pass—strings of $4 = 2^2$.

ST S6 S4 S3 H9 H3 DJ CQ, SK S7 HT H8 H2 DQ DT D9, S9 H7 D6 CA C8 C4 C3 C2, SA HQ H4 D7
SQ HK HJ H5 D5 D2 CK C6, SJ S2 HA D3 CT C9 C7 C5, S8 S5 H6 DA DK D8 D4 CJ,

(*d*) After third pass—strings of $8 = 2^3$.

SQ ST S6 S4 S3 HK HJ H9 H5 H3 DJ D5 D2 CK CQ C6, S9 S8 S5 H7 H6 DA DK D8 D6 D4 CA CJ C8 C4 C3 C2
SK SJ S7 S2 HA HT H8 H2 DQ DT D9 D3 CT C9 C7 C5, SA HQ H4 D7

(*e*) After fourth pass—strings of $16 = 2^4$.

SK SQ SJ ST S7 S6 S4 S3 S2 HA HK HJ HT H9 H8 H5 H3 H2 DQ DJ DT D9 D5 D3 D2 CK CQ CT C9 C7 C6 C5
SA S9 S8 S5 HQ H7 H6 H4 DA DK D8 D7 D6 D4 CA CJ C8 C4 C3 C2

(*f*) After fifth pass—strings of $32 = 2^5$.

SA SK SQ SJ ST S9 S8 S7 S6 S5 S4 S3 S2 HA HK HQ HJ HT H9 H8 H7 H6 H5 H4 H3 H2 DA DK etc.

(*g*) After sixth and final pass—strings of $52 < 2^6$.

FIG. 21.13. Merge sorting, planned string length.

each half is considered. The spade four (S4) is smaller than the spade ten (ST), so the spade ten is recorded first on the first output tape [Fig. 21.13(*b*)]. Then the second cards (club queen and diamond jack) are compared and recorded on the second output tape, the DJ going first because it is the larger. Then the H9 and H3 go on the first tape, etc. Now each tape has only 13 strings on it. The first cards on each tape (ST and DJ) are compared; the larger (ST) is recorded on the first output; the S4 which followed the ST is compared with the DJ and, being larger, is recorded following the ST; there being no more cards in the ST, S4 string, the DJ and CQ are chosen next by default. The process is repeated in putting the S6, S3, H9, H3 on the second tape, and again in putting the HJ, H5, CK, C6 on the first tape, and so on.

Since each pass over the tape halves the number of strings, one quickly sees that n passes will arrange 2^n items in sequence by this planned-string-length merge-sorting procedure. Thus x items will require $\log_2 x$, or $3.3 \log_{10} x$ ($\log_{10}$ being the common logarithm found in tables) passes. For example, 750,000 items require $3.3 \times 5.875 = 19.39$ or (since one cannot make a partial pass) 20 passes to arrange in sequence by this procedure.

Variable-string-length Merge Sorting. In the planned-string-length procedure just described, the end of each input string was determined by counting the items and stopping when that number reached the planned length of 1, 2, 4, 8, etc., respectively,

in the successive passes. This procedure makes no use of any ordering that the inputs may initially have—either by chance or by the nature of the source data which may be partly in sequence when it originates.

Even in a completely chance arrangement of a group of items, such as the initial arrangement of the playing cards in Fig. 21.7, some of the items will fall into sequenced strings of two, three, or more items. Figure 21.14(*a*), for example, shows the same deck of cards—with commas separating the strings which were formed by chance. In fact, the average string length (number of items divided by number of strings) in a long random sequence appears to be 1.86 items per string. (The average length of the string on which a randomly selected item falls is $2e - 3 = 2.44$, according to H. H. Seward, but this figure is not particularly useful in comparing the variable-length with the planned-length procedure. The 1.86 average is not very satisfactory either, since what is needed is a distribution, not an average; but it gives some measure of likelihood, and greater refinement is hardly worthwhile.)

S4 CQ, H3, S3 CK, H5, SQ D2, D9, H2 DT, S7, SJ C7 C5, S2 H7 D6 C2, C4, CJ, D4, DA, S5, HQ, SA
ST DJ, H9, S6 C6, HJ D5, HK DQ, HT H8, SK D3 CT, HA C9 C3, S9 CA C8, H6 D8, DK, S8 D7, H4

(*a*) Initial arrangement, divided in half—strings of at least $1 = 2^0$. (Note that this chance arrangement falls into 12 strings of 1, 12 strings of 2, 4 of 3, and 1 of 4.)

ST S4 DJ CQ, S6 S3 CK C6, SQ HK DQ D2, SK H2 DT D3 CT, SJ S9 CA C8 C7 C5, DK C4, H4 D4, S5 HQ
H9 H3, HJ H5 D5, HT H8 D9, S7 HA C9 C3, S2 H7 H6 D8 D6 C2, S8 D7 CJ, DA, SA

(*b*) After first pass—strings of at least* $2 = 2^1$.

ST S4 H9 H3 DJ CQ, SQ HK HT H8 DQ D9 D2, SJ S9 S2 H7 H6 D8 D6 CA C8 C7 C5 C2, H4 DA D4
S6 S3 HJ H5 D5 CK C6, SK S7 HA H2 DT D3 CT C9 C3, S8 DK D7 CJ C4, SA S5 HQ

(*c*) After second pass—strings of at least* $4 = 2^2$.

ST S6 S4 S3 HJ H9 H5 H3 DJ D5 CK CQ C6, SJ S9 S8 S2 H7 H6 DK D8 D7 D6 CA CJ C8 C7 C5 C4 C2
SK SQ S7 HA HK HT H8 H2 DQ DT D9 D3 D2 CT C9 C3, SA S5 HQ H4 DA D4

(*d*) After third pass—strings of at least* $8 = 2^3$.

SK SQ ST S7 S6 S4 S3 HA HK HJ HT H9 H8 H5 H3 H2 DQ DJ DT D9 D5 D3 D2 CK CQ CT C9 C6 C3
SA SJ S9 S8 S5 S2 HQ H7 H6 H4 DA DK D8 D7 D6 D4 CA CJ C8 C7 C5 C4 C2

(*e*) After fourth pass—strings of at least* $16 = 2^4$.

SA SK SQ SJ ST S9 S8 S7 S6 S5 S4 S3 S2 HA HK HQ HJ HT H9 H8 H7 H6 H5 H4 H3 H2 DA DK etc.

(*f*) After fifth and (in this case) final pass—strings of at least* $32 = 2^5$. (Note that if there were one more string in the first half in the initial arrangement—if, for example, the S5 and HQ near the end were interchanged—a sixth pass would be required.)

* Except possibly the end strings when one "half" has more strings than the other half.

FIG. 21.14. Merge sorting, variable string length.

Figure 21.14 shows the same deck as in Fig. 21.13 sorted by the variable-length-string method. In the example deck, there are only 29 strings, and not 52 as there might possibly be and as is assumed in the planned-length procedure described earlier. The average string length in the example is $52/29 = 1.79$. Since 29 strings should require only $\log_2 29 = 4.85$ passes, five passes should and do suffice in this case using the variable-length procedure, compared with the six required regardless of arrangement using the planned-length procedure.

Notice, however, that for optimum results the deck should initially have been divided into halves having an equal number of strings, not an equal number of cards. The number of passes depends on the greater of the number of strings in each half. In the example given (a chance arrangement in which fate was given a little nudge), a sixth pass would be required if there were even one more string in the first half. If, for example, the S5 and HQ near the end were interchanged, splitting one string into two, the SA would, at the end of the second, third, and fourth passes, appear on the end of the first line and would form a second line all its own in the fifth pass, requiring a sixth pass to merge it with the rest of the deck.

Internal (Interchange) Sorting. Both digital and merge sorting are techniques for dealing with each card in a deck or each item on a tape in turn, disposing of it according

to some rule, and moving to the next one. Sorting in a series of passes this way is required by the sequential nature of the medium. In a random-access memory, however, some items can efficiently be tested and moved if necessary several times while others are left untouched. Either digital or merge sorting can of course be used in a random-access memory but they generally require more memory space than the so-called internal sorting procedures.

There are any number of variations of internal sorting by interchange, such as sorting by repeatedly finding the smallest (or largest) remaining item; by interchanging successive pairs if (and only if) the first is larger than the second; by "sifting" or by "partial sorting." The interchange sorting techniques all require time proportional to the square of the number of items, whereas merge or digital sorting require only time proportional to $x \log x$. Consequently, special variations of merge or digital sorting (requiring at most twice the storage capacity needed to contain all the items) are often preferable for internal sorting when speed is essential and enough storage capacity is available.

When relatively long items are to be sorted internally, time can be saved if the entire item is not shuffled about in storage throughout the sorting procedure. This can be achieved, at some sacrifice in storage space, by storing each item once and for all in some set of storage locations with some definite starting address and forming a set of pseudo items, one for each of the real items to be sorted. Each short pseudo item, consisting only of the key and the address of the start of the real item which it represents, is then put through the desired internal sort. The result is a list of the addresses of the items, rather than the items themselves, arranged in sequence according to the item key. Then each item can be processed or recorded on tape in proper sequence by using each address in sequence to select the proper item next.

The biggest use of internal sorting is as a preliminary to merge sorting on tapes. The data are sorted internally into initial strings of one or more tape blocks in length, permitting everything to be handled in block-sized chunks thereafter. Some drum- or disk-type memories can be searched very rapidly for the largest key present. Such special features lend extra attractiveness to internal sorting techniques.

21.2.10. Choice of Sorting Techniques. There are, as shown above, three basic types of sorting techniques—digital, merge, and interchange. The choice of basic technique and of variation of that technique depends on a careful analysis of the problem at hand in terms of the total cost, including the cost of any delays in obtaining results occasioned by choosing a slow technique.

Given w different items, each with an average of d digits, with a key word (condensed or not) of r decimal digits in range (i.e., the $\log_{10}$ of the difference between the decimal equivalent of the largest representable key and that of the smallest), to be sorted using $2b$ tape units capable of being processed at a rate of s digits per second: the time required to perform the tape sort is, where brackets indicate "the largest integer less than the enclosed quantity,"

By merge sorting: $$\frac{wd}{s}\left(\frac{\log w}{\log b} + 1\right) \quad \text{sec}$$

By digital sorting: $$\frac{wd}{s}\left(\frac{\log r}{\log b} + 1\right) \quad \text{sec}$$

Internal sorting by interchange, given a random-access storage system and a computer that can compare keys and interchange two d-digit items at a rate of t interchanges per second, requires

$$\frac{w^2}{2t} \quad \text{sec}$$

As a general rule, punched cards are sorted digitally while magnetic tapes are merge-sorted. In the rare cases where the range r of the key is less than the number w of items (implying that on the average more than one item bears a given key), a digital-tape sort may be in order. Internal sorting is used on small volumes of data, especially as a preliminary to merge sorting of tapes. No such preliminary to digital sorting appears practical at present.

21.3. MANAGEMENT PROBLEMS*

21.3.1. The Problem Defined. As discussed in the introduction to Art. 21.2.1 the subject of data processing can include not only clerical operations in a business but also the handling of data relative to management decisions. The present section will be concerned with the application of electronic computers to the decision-making and policy-making operations of a business.

It should be emphasized at the outset that this discussion is not concerned with "push-button management." This is not a question of an electronic computer sitting in the president's chair and directing the efforts of the people in the company. Rather, the computer is considered to be a *tool*, to be used by management for making more optimum decisions. And it is expected that the requirements for a good manager will become more exacting as these new tools are put into use.

Also, it should be emphasized that this section does not advocate that electronic computers be used for all management decisions. Rather, it hopes to point out some types of management decisions in which computers can be of assistance.

Throughout the discussion, an attempt will be made to point out *how* electronic computers can be applied to certain management problems, and to cite examples of actual applications. At the same time, the role of human judgment will be pointed out to show the aspects of the problems which do not appear to be easily mechanizable.

Major Aspects. There are a number of ways in which computers have application to management problems, including the following areas:

1. Management strategy. This area is concerned with long-range planning of business operations and the establishment of policies. Computers can be of assistance to management by working out the consequences and probabilities of occurrence of various alternative courses of action. Management is then in a position to compare the various consequences and to select the alternative that gives the most desirable consequence.

2. Management tactics. This area is concerned with the more immediate, localized decisions, which are in turn governed by the strategies that have been adopted. Computers can be of assistance to management by working out the details of any particular decision—such as the optimum stock level for a particular item carried in stock, considering the most up-to-date information on demand for the item, availability of supply, and so on.

This area of management tactics is perhaps the one most subject to "mechanized decision making" where the data-processing system can take over the functions previously performed by order clerks. An example will be cited later of system plans which call for almost completely automatic handling of reordering decisions, in a military-supply system.

3. Management control. This area is concerned with actually directing and controlling the operations of the business, once the tactical decisions mentioned above have been made. This area includes clerical data-processing operations, for issuing the necessary orders, maintaining the files, and issuing the management reports described in Sec. 21.2. It also includes the actual operations of the firm—producing goods (in a manufacturing plant), selling goods (in a store), moving goods (in a transportation company), and so on. This is the area generally given the name "automation," and much of it is described in Sec. 21.5.

Each of these aspects of management strategy, management tactics, and management control is a problem because management is constantly faced with the necessity of choosing the optimum course of action. The introduction of computers into the system can be of value, for a number of reasons. For one thing, machines can usually perform more economically than can humans, for those aspects of an operation which can be mechanized. Also, the machines are less variable than are humans, so that their performance can be predicted more accurately.

* By Richard G. Canning and Roger L. Sisson.

The Integrated-system Concept. The application of electronic computers to management problems can go beyond just the three individual major areas mentioned above. It seems very possible (and actual examples can already be cited) for computers to be applied to *combinations* of those areas, to obtain still more optimum results.

For instance, it seems quite possible to design a computing and data-processing system that performs the tactical decision making, develops the detailed instructions based on those decisions, transmits the instructions to the automated plant where the instructions are performed automatically, records the actual performance and posts this performance in the master records, and then summarizes the performance for the next tactical decision-making operation. Under such a system, it becomes difficult to draw a clear line between the areas of decision making, clerical processing of data, and the automatic control of production.

Two Levels of Application. Not only can electronic computers be applied to the different areas described above but they can also be applied at two different levels:

1. System design. This level is concerned with the laying out of an operating system. It involves mostly theoretical work (using a computer as an analytical tool) which leads to a practical solution of day-to-day management problems.

2. System operation. This level is concerned with the actual day-to-day operation of the system, in which the computer is one cog in the system. In this case, the computer is an operating tool rather than an analytical tool.

These, then, are the different areas and the different levels within those areas where electronic computers can be applied to management problems. The discussion which follows will describe how computers can be applied to each of these and will cite examples of what is being done today.

21.3.2. The State of the Art. *Problem Formulation.* The electronic computer, as it is delivered by the manufacturer, is basically "empty." While it has the capacity to solve complicated problems, it must be given the exact method to be used for each problem, in great detail. This fact has certainly been made clear in previous sections of this handbook.

There is need, therefore, for the human operators of the computer to develop a step-by-step procedure which the machine must follow in solving the problem. And before such a procedure can be developed, the problem must be stated explicitly. The physical problem—the problem facing management—must be stated in symbolic terms. And this symbolic statement of the problem must be sufficiently realistic (similar to the real physical problem) so that the answers will be accurate enough to be usable. If management's problem has many interrelated variables that cannot be ignored, then the symbolic representation should include those variables, and with their proper relationships. If uncertainty is involved, where the values of some of the variables cannot be predicted accurately by management (how many electric fans will be sold next summer—which in turn depends on how hot it will be next summer), then the symbolic representation must include uncertainty.

The development of a suitable symbolic representation of the problem, and a step-by-step procedure for solving the problem, fall into the category of "problem formulation." This area is still the responsibility of the human beings; machines have made no dent in it as yet.

The use of the scientific method in problem formulation has been given the name "operations research" in recent years. The results of this scientific approach have been very encouraging and indicate that important improvements can be made in management methods. Considerable use of mathematics has normally been made in operations-research studies because of the large number of variables involved and their rather complex relationships. Thus the symbolic representation of the problem is often in the form of a mathematical model.

The mathematical model which is developed through this problem formulation can take on a wide variety of forms, depending upon the characteristics of the particular problem. There are certain forms, however, that seem to occur more frequently than the rest, in that they tend to represent the problems that are common to wide varieties of businesses.

One of the more common forms of management-problem representation is by the use of large matrices. This formulation is particularly useful in product-mix problems, where management is attempting to determine how much of each of several products to order, so as to maximize profit or minimize cost and still stay within a working-capital budget and other budgets. The number of products may reach up into the hundreds, while the number of equations relating the unknowns often number from 50 to 100, sometimes more in the larger problems.

Another form of problem representation that is often used involves the aspect of uncertainty. Take the hypothetical example where the exact number of fans that will be sold next summer is not known. But from a study of the weather in past years, and the number of fans sold, a distribution might be set up which says, in effect: the probability that X or more fans can be sold is 1 chance in 10; the probability that Y or more fans can be sold is 2 chances in 10, and so on. Then the other variables involved, such as cost of production, distribution costs, and inventory costs, are included in the model. In the solution, random numbers are chosen to determine what the demand for fans will be next summer, and from this the profit can be determined for different production schedules. Then a different set of random numbers is selected, and the problem is solved again. After the process has been repeated numerous times, the results may tend to indicate a desired course of action, which might be that production schedule which will maximize possible profit, or one that will minimize possible loss.

"Game theory" is a term which deserves mention, also. In a way, it might be considered a philosophy of thinking more than a specific type of problem formulation. That is, the use of large matrices, distributions, random numbers, and other types of problem formulation will often be found in the game approach. Game theory is concerned with optimum strategies—trying to out-think the opposition. Its major elements are attempting to maximize the minimum possible gain, or to minimize the maximum possible loss, in a course of action, and also the element of systematically introducing random behavior to make it more difficult for the opponent to predict behavior.

There are a number of other common forms of problem representation which could be similarly described, but it is outside the scope of this section to give this detail. One that is worth mentioning is "waiting-line theory," or queuing theory, where the effect of random arrivals at a toll booth (for instance) is studied. Some important problems are combinatorial in nature, such as the assignment of jobs to machine tools in a job shop; it is often the case that the total number of possible combinations is tremendous, exceeding the number of molecules in the universe, and it is desired to select the optimum combination out of this large number. Present-day mathematical techniques do not even hope to obtain the optimum in such problems but rather are satisfied with obtaining "better" solutions than the rule-of-thumb methods so often used by management.

No further discussion of the types of problem formulation is warranted here. For the interested reader, a number of excellent references are listed at the end of this section.

A few final comments are in order, however, It will be noticed that the word "optimum" occurred several times in the above discussion. The methods attempt to obtain the optimum answer. Someone must define what an optimum is, in each case—whether it is the maximizing of profits, the minimizing of possible losses, or something else. This is the role of management judgment, when using these mathematical techniques; this is another part of the problem (together with problem formulation) that has not yet been mechanized. Another point should be made, also. In the solution of large matrices, an electronic computer is practically mandatory because of the tremendous number of routine calculations involved. Similarly, when many alternative production schedules are analyzed, for different levels of demand, electronic computers are practically an economic necessity in many cases. So electronic computers can be expected to play an important part in aiding in the solution of management problems in the future.

Again, a word of caution is in order. Much of the early work in operations research has resulted in problem formulation that has not required electronic computers for

solution. In fact, the use of electronic computers is more the exception than the rule, to date. The reason seems to be that the more localized problems have been chosen by the operations researchers, in order to produce tangible results in a short time. As the more general management problems are tackled, and especially in the larger firms, then the use of computers will become more extensive.

The application of such types of problem formulation, particularly those involving the use of electronic computers, to management problems will now be discussed, to indicate what has been accomplished. The application to systems design will be discussed first, followed by applications to systems operation.

Systems Design. As mentioned previously, this level of application is concerned with the laying out of an operating system and developing a set of optimum rules to follow. An electronic computer is used as an analytical tool, for analyzing consequences of alternative rules.

1. Management strategy. A good example of work in this area was done by the Stanford Research Institute and reported at the 1955 Operations Research Convention in Los Angeles. It involved strategic inventory decisions for a railroad supply system. Available historical data were gathered for several years in the past, indicating the number of orders placed, the valuation of inventory, the number of times requisitions could not be filled because of out-of-stock conditions, and so on. Based on these data, and using an electronic computer, families of curves were developed which related number of purchase orders placed with inventory valuation with out-of-stock conditions. With these curves, management would be able to set a maximum desired inventory valuation (based on available working capital) and a maximum on the percentage of out-of-stock cases (based on judgment and experience). When these two strategic decisions were made, the number of purchase orders to be placed annually was thereby determined. And knowing the number of purchase orders and the annual requirements for parts, the quantity per purchase order could be determined. All the tactical decisions on what to order, when to order it, and in what quantity were then automatically established—just by making the two important strategic decisions. Of course, a data-processing system would be needed for actually handling the detail purchase orders, requisitions, determining changes in requirements, etc., but the system as a whole would be expected to work within the limits specified by the two strategic decisions—maximum inventory valuation and maximum allowable out-of-stock conditions.

This area of application, systems design for strategic decisions, has not yet received too much attention; the example mentioned above is one of the first. It is an area where game theory may possibly be applied, where the consequences of several alternative courses of action are developed and management selects that course of action which has the best over-all consequence.

2. Management tactics. This application is concerned with developing the best set of rules to follow in making the day-by-day operating decisions. In the case of the railroad inventory system mentioned above, the tactical decisions are practically determined when the strategic decisions are made.

An example of the use of electronic computers in developing tactical-decision rules can be found in the work of the Management Sciences Research Project, University of California, Los Angeles. The work is concerned with shop scheduling in a job shop—determining when each operation of each order in a machine shop should be started, so as to complete all the orders as close to due date as possible. In developing the rules, a program for a digital computer was prepared which simulated the movement of orders through a machine shop. Different memory cells in the computer were used to represent different machine tools, for instance, and jobs were assigned to these machine tools by storing the job number in the proper machine-tool memory cell. Then different decision rules were tested for selecting the next job for a machine tool when several jobs were waiting for the same machine. The objective, of course, was to find the rule which most efficiently put the jobs through the shop on schedule.

Another example of the use of digital computers for developing tactical-decision systems has occurred at the University of Michigan, in connection with traffic studies.

The problem is to determine the best rules for controlling intersection traffic, where some cars want to make right turns, some left turns, some straight ahead, and there are pedestrians present. As in the case of shop scheduling mentioned above, the computer program was prepared which simulated the intersection conditions. Then "cars" were fed into the intersection at random (but at a rate consistent with actual traffic conditions). Likewise, the cars which desired to make left and right turns were also selected at random, consistent with actual statistics. The procedure in this case was to explore the effect of different traffic rules and their relative efficiency in getting the cars through the intersection.

Several comments are appropriate. Both the examples mentioned above used the technique of simulation, where the physical situation is simulated by a computer program. Then the computer is used to play a game, tracing out the consequences of different rules. In both the above cases, the original "model" was quite simple, and it was desired to find a method that was in the "right ball park." Then, as results were obtained which looked more and more satisfactory, the model was expanded to take more factors into consideration.

It has often been stated that analog computers are most appropriate for problems of simulation; certainly that is an area where they have been doing excellent work. The examples mentioned herein, however, indicate a type of simulation where analog machines may not be desirable—namely, where the physical situation consists of discrete units such as shop orders or automobiles. This is just another way of saying that the characteristics of the problem determine which type of computer is best suited for its solution and that it is not wise to generalize about the merits of a particular type of machine.

3. Management control. This application is concerned with using electronic computers for developing optimum rules for management control. That is, once the tactical decisions have been made by management, orders are issued to the organization to carry out those decisions. Management control is needed to ensure that the organization does, in fact, carry out those decisions or, conversely, if the decisions are not carried out properly, to detect that fact so that management can issue corrective-action orders.

Work in this area of optimum control-systems design for business systems is still mainly in the research stage.

Some pioneer work in problem formulation has been carried on by Professor Donald P. Campbell at MIT.[21] In his article on "Dynamic Behavior of Linear Production Systems," Professor Campbell applies servomechanism theory to a production system. He uses mathematical techniques for developing certain optimum rules for management to follow. For a more complex formulation, where more of the variables are included, it might be desirable to set the system up on a computer for ease of solution. It might be mentioned that the problem, as Professor Campbell formulated it in his paper, appears quite suitable for analog-computer application.

Some other work along this line is being done at the Management Sciences Research Project, University of California, Los Angeles. Again, a digital-computer program is being used to simulate a machine-shop condition where work is being performed by a machine-tool operator in accordance with orders he receives. Such factors as the need to rework parts and the scrapping of incorrectly made parts are included in the program; random numbers are used to determine which orders require rework or scrapping. Deviations of completion dates from due dates are being studied, for different values of the load on the machine tool (in terms of its capacity).

4. Integrated systems. This application is concerned with using electronic computers for developing optimum rules of operation for practically the entire business enterprise—decision making, the automated operation, and control.

Systems Operation. This level of application is concerned with the actual use of an electronic computer in the everyday operation of the business. As such, it is a direct replacement for certain types of clerical work and supervisory work.

1. Management strategy. Since the field of operations research is so relatively new, it is to be expected that electronic computers are not yet widely used by management for aiding in making strategic and policy decisions.

One possible application which might fall into this area is that of preparing for collective bargaining. When management (or the union) is preparing for negotiations, it would be of considerable assistance to have the consequences of the different demands worked out in a fair degree of detail. The expected demand for the product for the next year could be expressed in terms of "uncertainty," described above under problem formulation. Then the profit position (or loss position) of the firm could be analyzed for different levels of pay and fringe benefits. Random numbers would be used for selecting the demand for the product in each trial. After the computer has made numerous trials a picture should develop which indicates the most desirable course of action.

There are numerous other applications that could be mentioned as possibilities. Most of them appear to be applications of game theory, which has been receiving considerable attention in recent years. Strategy in a competitive market will certainly force increased use of electronic computers in the years to come.

2. Management tactics. It is in this area of using electronic computers to aid in the everyday tactical decisions of a business that the most attention probably has been paid.

One of the more common applications centers around inventory control, where it is desired to minimize costs of carrying inventory for each item stocked and still have only a small percentage of out-of-stock conditions. The most common approach is to develop an economic-order-quantity formula, which specifies how much of any given item is to be ordered, taking into account cost of ordering, inventory-carrying costs, cost of obsolescence, expected demand, and sometimes lead time. (The equivalent of this approach under past management methods is the phrase "We always try to carry a three month's supply on hand.")

The values of the various parameters just mentioned must be determined for each item stocked. This in itself involves a large clerical job. But even above this, the method must be made dynamic, so that the values of the parameters are kept up to date, reflecting changes in customer demand, procurement time, etc. In such a case, an electronic computer will show many advantages over manual clerical systems and punched-card systems. As items are requisitioned, the computer will compare the actual demand against the original predicted demand and (based on the method prescribed for it) will adjust the reorder points, and reorder quantities to reflect the actual demands. Thus the computer will perform the functions of deciding when to reorder (when the on-hand balance reaches the reorder point, which the computer has calculated) and how much to reorder (from the reorder quantity, which the computer has calculated). An actual system planned, with this concept in mind, will be discussed below.

Another example of mechanizing tactical decisions that has been studied in considerable detail involves the scheduling of steel production. The problem is basically one of selecting an optimum product mix in order to meet customer demand: Given several types of coal, with varying costs, how to combine them in the coking ovens to produce the desired cokes at the least cost. Given the cokes and several varieties of iron ore which are on hand (and the costs of each), how to combine them in the blast furnaces to obtain the pig irons desired at the least cost. Given the pig irons and available steel scrap (each with various constituents and costs), how to combine them in the open-hearth furnaces to obtain the steels desired. Given the steels and the rolling-mill capacities, how to produce the steels to meet the customer orders. These are decisions that steel-plant management is faced with day after day. No one final decision can be made, because the conditions are constantly changing—the costs of the ores which can be obtained, the orders for different types of steel by the customers, etc.

As discussed under Problem Formulation, this steel-production problem involves the solution of large matrices, when being performed on an electronic computer. Present research work is concerned with only limited aspects of the total problem described above, because of the size of the total problem as well as the fact that certain aspects (such as coke production) are very nonlinear in nature and are not easily handled by present computing techniques. However, it is a good example of work

that is being done where an electronic computer aids in the day-to-day tactical decisions.

The shop scheduling operation, which was mentioned briefly under systems design for tactical decisions, is another example where an electronic computer can aid in the actual systems operation. In the previous systems-design discussion, the role of the computer was to aid in selecting the optimum rules for the shop to follow. In systems operation, those rules must be used day in and day out. Research on this problem, at the Management Sciences Research Project, indicates that a computer may be used to generate the shop schedule once a week. Each week, it generates the shop schedule for the next two weeks, beginning with the latest reported condition in the shop.

Other examples of production scheduling have been discussed by various people throughout the country. In an aircraft plant, when bidding on new aircraft contracts, the month-by-month labor requirements for each of several possible delivery schedules are computed. The computation takes into account the present contract commitments. The objective is to find the delivery schedule which can be bid that delivers the aircraft to the customer at the fastest rate and still does not require any sudden peaks in the labor load. While the volume of such calculations is not great (a few a week), the time saving and more thorough analysis are important. In another example, an assembly line is used for producing numerous types of assemblies during the month—the volume on any one type not being great enough to keep the line busy for the full month. The problem is to assign jobs to the assembly people down the line; so that jobs are performed in the required sequence, each person has a full 8 hr of work a day, and the number of people on the line is minimized. Such job assignments have to be recomputed each time a new product is introduced or a significant change made on an old product.

The area of tactical decisions in management, then, is one where electronic computers can be expected to play a more important part in the future.

3. Management control. The discussion of the application of electronic computers to management-control operations is considered in two other sections, Secs. 21.2 and 21.5. The reader is referred to those sections for further details. Numerous installations have already been made in these areas, and more are being installed constantly.

Some general comments are in order, however. The connection between data processing and management control may not be readily apparent to the reader. Data processing is usually associated with clerical activities—such as issuing orders, premium notices, and pay checks, maintaining files, and preparing reports. A little thought will reveal that files are maintained and reports prepared primarily for management-control purposes, and in some cases governmental-control purposes. Management wants to know what was actually done; so that corrective action can be instituted—or pay checks issued. So the application of electronic computers to the data-processing (clerical) operations in a business really is concerned with management control of that business.

Sometimes, the volume of one type of transaction within a business is sufficient to warrant an electronic system, and the remainder of the data-processing system within the firm might remain manual or punched-card. An example of this is the reservation problem for airlines and trains. It is desired to control the number of seats sold on each flight—so as not to oversell a flight, or conversely, not to turn a customer away when seats are actually available on the flight of his choice. A system has been installed and working, in New York City, for several years which handles peak loads at 600 inquiries per hour. Other important data-processing operations within the firm are not handled by this electronic system, such as keeping track of the number of flight hours for each crew member, each aircraft, and each engine—or the rescheduling of crews and equipment when certain flights have been canceled or delayed by weather.

In most types of business, the volume of any one kind of transaction may not be sufficient to pay for the cost of the electronic system (rented or purchased). In such a case, the electronic system must be time-shared between a number of data-processing

operations. A good example of this was the Univac installation at the General Electric Company, Louisville, Ky. The original plans called for the application of the computer to four major data-processing operations: payroll, material control, order service, and general and cost accounting. These operations by no means constituted the whole data-processing needs of the plant, but they were important areas suitable for mechanization.

4. Integrated systems. Completely integrated systems involving electronic computers are still in their infancy. There are two systems of interest that begin to approach the completely integrated system.

The Kidde Warehouse System, designed by Walter Kidde Constructors and Teleregister Corporation, has been built as a prototype at the Link-Belt Company. Incoming orders are broken down by the items requested, and a punched card is prepared for each item (including customer name, address, etc.). The cards are sorted by warehouse location and are sent to the stock pickers. Cards for many customer orders go to each picker. The picker obtains the merchandise and places it, with its card, in a traveling overhead tray. A dispatcher takes the card from the tray, puts it in a reader, and two actions take place: (1) the merchandise is directed to unload at a chute corresponding to the customer's number, (2) a count is made by that customer's number. When all the customer's packages have been collected, a signal tells a packer that the order is ready for packing. Here is a system, then, that begins to combine data processing and machine control. The orders are issued to the pickers in a form that can be recognized by human beings as well as by machines. Eventually, it should be possible to have much of the picking done automatically, from orders issued by the data-processing system.

Plans for another system have been developed by the Rand Corporation for the United States Air Force. This system combines decision-making and data-processing operations for a military supply system. As mentioned above under tactical decisions, an electronic computer can keep track of the actual demand for an item and can recompute the reorder point and reorder quantity for each item from time to time, so as to minimize inventory costs. In addition, the electronic system can issue the actual purchase orders themselves, subject only to review by the purchasing agent. Also, the electronic system can maintain the perpetual inventory on all the items in stock, maintaining the inventory files as items are ordered, received, issued, and returned. It is from these actual transaction data that the new calculations can be made. Here, then, is an example of a system that combines the data-processing and decision-making operations of a business.

21.3.3. Economic Considerations. With regard to the economics of using electronic computers to aid in the solution of management problems, two major questions might be asked: From the experience of the pioneering companies, what is the order of magnitude of the costs involved? And how can these costs be paid for? Each of these questions will be discussed briefly.

What Are the Costs Involved? There are four major cost areas to be considered: problem formulation, programming and debugging the program, "installation," and operating costs.

1. Problem-formulation costs. This cost is extremely difficult to estimate, since it depends so greatly on the complexity of the problem and the method of approach. Perhaps the simplest approach is to use practically the same method that is now being done manually, but programmed for an electronic computer. In the example described above (Art. 21.3.2) for computing an acceptable aircraft delivery schedule, for bidding, this approach was followed. The method used on the electronic computer was very similar to that already used by scheduling experts in the plant but with a few rather obvious improvements.

Another example required a complete departure from present methods. The formulation, programming, and running of trial problems required only about 3 man-months. Total cost was in the $2,000 to $4,000 range.

At the other end of the range are the problems that require 1 to 2 man-years for formulation, before programming even begins. The steel-mill problem is an example of such a case. Of course, once the problem has been formulated by someone, and published (if possible), subsequent users should require much less time for formulating similar problems.

As a rule of thumb, it would be well to estimate at least 6 man-months for the formulation of sizable management problems, unless there is prior information to the contrary. Certainly, a manager's bland statement that "This shouldn't be a tough problem to formulate; we're solving it all the time" is not a good yardstick, if a true operations-research approach is to be followed. As the operations researcher gets into the problem, it is possible he will find many more important variables, with complex relationships, than was first apparent from the manager's description.

A word about future machine design is warranted. There should be a much closer coordination between problem formulation and machine design than there has been in the past, at least for certain major classes of problems. Perhaps the problem of predicting sales in a department store is a good case in point. Problem formulation may call for the measurement of certain merchandise characteristics that are not now measured—silhouette, combination of colors, relative magnitude of color areas, etc. Also, problem formulation may call for rather complex relations involving these characteristics. It is up to the machine designer to develop a machine capable of the efficient solution of the problem. While the present-day general-purpose computer probably can be programmed to solve any such problem, it is still not evident that such a machine will meet the economics of the situation.

2. Programming and debugging the program. These costs are also nontrivial. It is hard to estimate the cost of programming because of the great differences in the abilities of programmers. An inexperienced programmer may require a month to do what an experienced programmer can do in 2 days—based on results to date with these machines.

Programs of some 4,000 instructions are not uncommon in problems of this type; in fact, the programs can be several times this length. Rule-of-thumb estimates indicate that an experienced programmer might average between 20 and 340 program steps per day, if he has to work out the details of solution, and might average several hundred instructions per day if the method of approach is quite clear in his mind. Programming a company's payroll operation for an electronic computer, as an example, appears to fall in the 20- to 40-instruction-per-day range, even though payroll has been done satisfactorily on punched-card machines, simply because the new arrangements of the master files requires new detail methods of processing.

Debugging the programs requires more time than is generally recognized. In some computer installations, about 50 per cent of the available computer time has been used for this operation, although techniques have been developed to cut down this time requirement. As a matter of fact, with long computer programs, involving some 6,000 to 10,000 instructions, it is often impossible to set up test problems that test all possible combinations in the program; so that the programmer is never really sure that his program is right. It is not unusual to find errors in the program after it has been in use for a year.

An encouraging development in the field, to reduce programming time and debugging time, is that of automatic programming. This technique can be used where the procedures to be used can be standardized. The programmer then need only refer to a set of specific standard procedures, and the computer will locate the programs for those procedures and link them together to form a running program. The Remington Rand A-2 compiler, for instance, has developed 4,000-instruction Univac programs from only 435 instructions specified by the programmer, and the compiling required only 20 min on the Univac. Moreover, if the compiler has been carefully debugged, then the resulting programs should be error-free and should require little or no debugging.

More recently a committee in the Department of Defense of the United States Government has been instrumental in establishing a *Common Business Oriented Language* called COBOL. In many respects this language is similar to ALGOL (see Art. 17.33). This language is to serve as a means of communicating computational algorithms (particularly those relating to business data processing), and for many

cases, problems written in this language may be run on any of a number of different computers (provided COBOL translators are available).

A word about machine memories is in order. The solution of product-mix problems will involve matrices, often quite large matrices. And other types of management problems will require many thousands of machine instructions. The tendency appears to be toward larger and larger internal memories—50,000 to 100,000 decimal numbers, or even more. And the more rapidly those digits are available, the less over-all computing time will be required. It would be well not to underestimate the size of machine memories for business applications.

3. Installation costs. "Installation" costs are involved, even though a company just rents time on a service bureau computer. The "installation," in this case, covers the change-over costs, when operating the electronic-computer method in parallel with the present method. It is not yet clear just what the order of magnitude of these change-over costs will be for management decision-making applications. For data-processing (clerical) applications, it appears that even the smaller systems will require expenditures in the order of $100,000 to $200,000 for installation and change-over. It would seem reasonable to expect that computers for decision-making applications should not require expenditures of quite this magnitude.

4. Operation costs. This cost is also hard to estimate. If time is rented on a service-bureau computer, then the standard hourly rental (covering machine operators) would be the major cost. If the problem requires a complete data-processing system tied in with the organization (such as in the case of shop scheduling described above), then the cost of this data processing must also be considered.

How Can These Costs Be Paid For? Generally, management expects that costs such as those just described will be paid back by direct savings. And it is normally expected that the amortization period will be rather short, in the order of 1 to 3 years.

Data-processing (clerical) applications generally can show a certain amount of manual clerical or punched-card savings, to offset the cost of the electronic system. But for management-decision applications, it is much more difficult to put a finger on direct savings. In the example concerning bidding on an aircraft contract (see page 21–39), the company was spending a certain amount of money to have such schedules computed by hand; this money could be used to offset the cost of the electronic-computer calculations. At the same time, the calculations made by the computer went into a greater amount of detail than did the manual calculations; so that it was not fair to compare the two methods on direct costs alone.

In certain other types of management problems, such as the shop scheduling problem, it is almost impossible to find direct clerical costs that can be used to offset the cost of the electronic system. Rather, it is "shadow" costs that are involved here—the cost to the company of *not* doing efficient shop scheduling. Part of these shadow costs can show up as waste, loss of business due to poor deliveries, and so on. In any case, these shadow costs are very difficult to estimate accurately. And it is likewise difficult to prove that improvements in profit position were due almost entirely to the electronic-computer methods.

Perhaps management will become willing to forego direct, tangible cost savings in these decision-making applications, in the face of improved operations. If the general manager of the plant finds that his decisions are becoming more efficient and that he does not have to put in so much "overtime" in his office to keep up with the decisions, then he may well look on the computer as a valuable tool, even if he cannot point to direct savings.

21.4. SIMULATION*

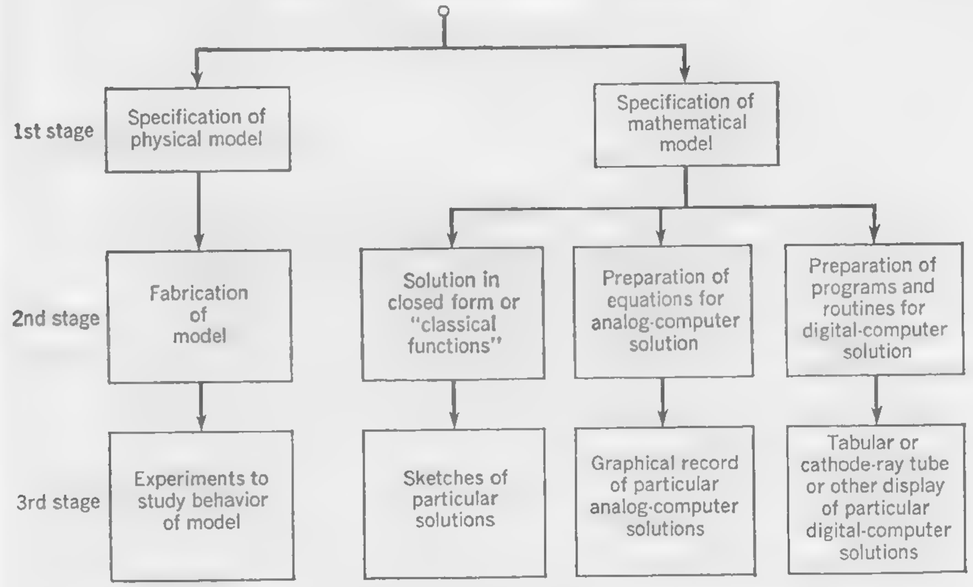
21.4.1. Principles of Simulation. The term "simulation" refers to the replacement of a given system by a substitute system, or "simulator," which responds to stimuli in a similar way and hence has the salient characteristics of the given system. The system may be a device, a process, or part of a device or process. The simulator is generally simpler, less expensive, and/or better adapted to experimentation than

* By Morris Rubinoff.

the original. Its purpose may be to expedite empirical study of specific system responses, or it may serve as an auxiliary adjunct or controlled environment in experiments on the original system or one of its parts. For example, an airplane simulator might be constructed to help predict the flight characteristics of an airplane still on the design board; on the other hand the flight simulator might be built as a flexible and safe device for inexpensive training of student pilots.

Simulation is achieved in three stages as illustrated in Table 21.1. The first stage consists of specifying the characteristics of the original system by means of either a physical or a mathematical model. The former calls for a set of detailed construction drawings while the latter usually takes the form of a set of algebraic or logical equations. In either case simplifying approximations are generally introduced in the

Table 21.1. Three Stages in Simulation


1st stage
Specification of physical model
Specification of mathematical model
2nd stage
Fabrication of model
Solution in closed form or "classical functions"
Preparation of equations for analog-computer solution
Preparation of programs and routines for digital-computer solution
3rd stage
Experiments to study behavior of model
Sketches of particular solutions
Graphical record of particular analog-computer solutions
Tabular or cathode-ray tube or other display of particular digital-computer solutions

representation of the original system, based on engineering intuition and subject to eventual justification by comparing simulator response with actual system behavior.

The second stage involves the preparation of the model for study. The *physical model* may be simply a copy of the original to a modified scale, or the scale may be distorted to emphasize the salient features under study. Preparation consists of physical fabrication. The *mathematical model* may have a solution in closed form, in which case simulation is not required. More often the solution is difficult to interpret or beyond the capabilities of existing mathematical techniques, in which case mechanical means are needed to obtain particular solutions. Preparation then consists of modifying the form of the mathematical model to render it suitable for solution either by analog or by digital computation, or by both. Mixed analog-digital techniques are particularly adapted to real-time simulation where part of the original system must be tied to the simulator.

The third stage of simulation is the experimental or computational determination of the responses of the simulator to families of input stimuli from various initial conditions. The responses may be obtained as motion film records of the physical model or as graphical records or tabular displays of computed behavior of the mathematical model.

21.4.2. Scope of the Section on Simulation. This section is concerned primarily with the digital or mixed analog-digital solution of mathematical models, i.e., with digital simulation. Preparation for digital solution of mathematical models requires one or more of (1) specifying the salient characteristics of the original system by means of functional relationships such as approximating polynomials or ratios of

polynomials, (2) expressing conditional relationships through sets of algebraic equations, (3) expressing system dynamic behavior by a set of ordinary or partial differential equations or by integral equations, and/or (4) defining states which call for discrete changes in simulator characteristics. Requirements (1), (3), and (4) will receive attention in this section.

Mixed analog-digital simulation employs encoding and decoding equipment for translation between the two number representations. Amplification, filtering, and other simulator operations can be mechanized by either digital or analog methods. The former is more versatile and less expensive, but it is generally better to use analog devices where the output is used to develop motion requiring a fair amount of power. Analog equipment is also used to represent man-operated continuous controls. These analog considerations are treated in the first part of the handbook and will not be discussed here.

The transmission of data between the analog and digital parts of a mixed system is generally accomplished by time-division multiplexing or sampling. This discontinuous but periodic exchange of data is best handled by sampled-data theory and Z transforms. These will be discussed below.

The advantages of digital simulation are (1) flexibility in alteration from one simulation to a second, (2) precision and accuracy to any desired extent, and (3) more sharply defined error analysis.

21.4.3. Function Approximation. The first requirement listed in Art. 21.4.2 for preparation of the mathematical model of a physical system was the specification of functional relationships between different parts of the system and its environment. These relationships are usually smooth single-valued functions of one or more variables, with at most a finite number of discontinuities. For example, wind-tunnel measurements of the aerodynamic coefficients appearing in airplane-flight equations show that these coefficients vary significantly with Mach number and angle of attack. The process of preparing these relationships for digital computation is referred to as "curve fitting" or function approximation.

Since polynomial approximation, either piecewise or over the entire range, is the most convenient form for digital-computer solution, attention will here be restricted to curve fitting by polynomials or ratios of polynomials. However, it should be kept in mind that other approximating functions (e.g., iterative computations) are also available and may occasionally be preferable for minimizing computing time and/or storage requirements. Similarly, although attention will be restricted to functions of a single variable, it should be kept in mind that functions of more than one variable can be represented using extensions of the methods described here.

The problem may be expressed more formally as follows: Given the single-valued functional relationship $f(x)$ defined over the range $x_0 \leq x \leq x_s$, to find a polynomial of degree n, or a set of such polynomials, which differ(s) from $f(x)$ by not more than $e(x)$. The error $e(x)$ may be a constant or a constant percentage of $f(x)$; it may be specified a priori as a function of x or it may be accepted on the basis of "engineering judgment" after $P_n(x)$ has been obtained.

Piecewise Approximation by Straight Lines. The simplest method of curve fitting uses a set of straight lines. The result is essentially a Chebyshev fit but it can be obtained by direct graphical construction. The procedure can also be programmed for digital computation, a primitive example of digital simulation of a graphical process. It is assumed that $e(x)$ has been specified in advance, as illustrated in Fig. 21.15.

The bounding functions $f(x) \pm e(x)$ are first drawn to scale. Beginning at one end of the range, say x_0, a straight line is drawn from x_0 to a second point x_1 on the same bounding function, tangent to the other bounding function and lying wholly between both. A second straight line is drawn in similar manner from x_1 to x_2, beginning on either bounding curve, whichever yields the larger value for $|x_2 - x_1|$. The process is repeated until the lines span the entire range x_0 to x_s. The approximating functions $P_i(x)$ then have the form

$$P_i(x) = m_i x + b_i \qquad x_i \leq x < x_{i+1} \qquad i = 0, 1, 2, 3, \ldots, k \qquad (21.\text{-})$$

as indicated by the solid lines in Fig. 21.15.

In the simulator, the particular $P_i(x)$ might be obtained by repeatedly halving the

range of x, in the manner of the method of bisection for determining the roots of a function. For example, if $k = 5$, the sequence of comparisons shown in Fig. 21.16 would be used. Note the test to ensure that x is within the prescribed range.

Alternatively, if x is known to vary slowly, the sequence might be to determine by comparisons first if x had increased or decreased, and second if the change were enough to call upon the succeeding linear segment.

Discontinuities such as that at x_2 in Fig. 21.15 will appear wherever $f(x)$ has an inflection. When such discontinuities are undesirable, the process for obtaining the set of approximating lines must be modified as shown by the dashed lines in Fig. 21.15.

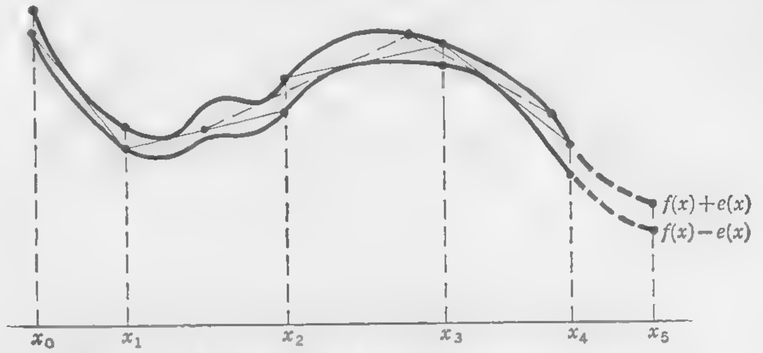


FIG. 22.15. Illustrating straight-line approximation.

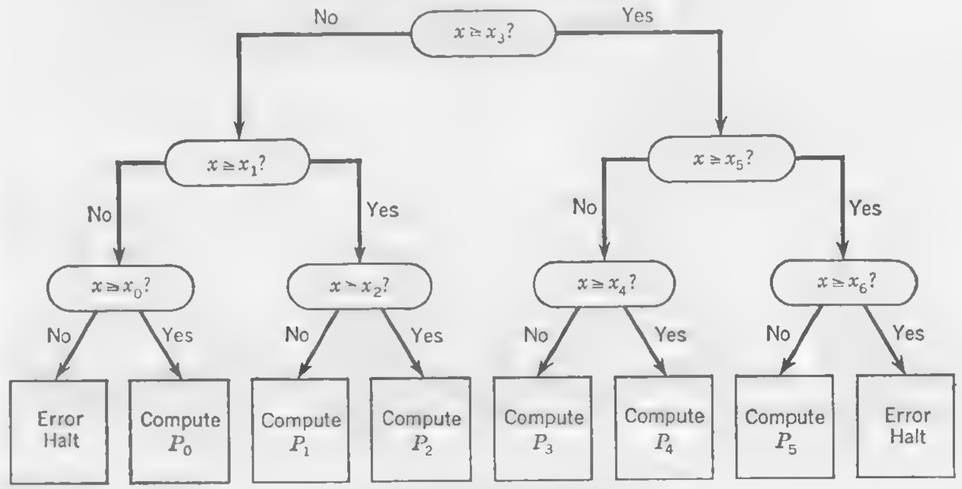


FIG. 21.16. Selection of correct $P_i(x)$.

Least-square Polynomial Approximation. This method is based on minimizing D^2, the square of the deviation of $f(x)$ from the approximating nth-degree polynomial $P_n(x) = \sum_{i=0}^{n} a_i x^i$. If $f(x)$ is given as a continuous function, D^2 is expressed in integral form; while if $f(x)$ is given only at discrete points x_k, D^2 is expressed as a sum. If closer approximation is desired over part of the range, a suitable weighting function $w(x)$ may be used as a multiplier. The two forms are then

$$D^2 = \int_{x_0}^{x_s} w(x)[P_n(x) - f(x)]^2\, dx \qquad (21.3)$$

or

$$D^2 = \sum_k w(x_k)[P_n(x_k) - f(x_k)]^2$$

In either case minimum D^2 depends on proper choice of the n coefficients a_i. These may be obtained by differentiating D^2 with respect to each of the coefficients in turn and setting the derivatives equal to zero. The resulting n equations are linear in the a_i and the latter are the only unknowns.

A major disadvantage of this procedure is that the coefficients depend upon the postulated degree of $P_n(x)$. Thus if the approximation obtained using a particular degree n_1 is unsatisfactory, the entire process including solution of a new set of linear equations needs to be repeated for a new polynomial of degree $n_2 > n_1$. A preferable method will now be described in which the degree of the approximating polynomial can be increased by simple addition of new terms.

The scale of the abscissa is first relabeled by means of the linear transformation

$$2x = (x_s + x_0) + u(x_s - x_0) \tag{21.4}$$

so that the new independent variable extends over the standardized range $-1 \geq u \geq 1$. The approximating polynomial is now expressed in the form

$$P_n(u) = \sum_{i=0}^{n} b_i p_i(u) \tag{21.5}$$

where $p_k(u) = \sqrt{(2k+1)/2}\, L_k(u)$ and $L_k(u)$ is the kth degree Legendre polynomial of the first kind. The latter may be defined by

$$L_k(u) = \frac{1}{2^k \cdot k!} \frac{d^k}{du^k} (u^2 - 1)^k$$

which yields for the first few polynomials

$$\begin{aligned}
L_0(u) &= 1 \\
L_1(u) &= u \\
L_2(u) &= \tfrac{1}{2}(3u^2 - 1) \\
L_3(u) &= \tfrac{1}{2}(5u^3 - 3u) \\
L_4(u) &= \tfrac{1}{8}(35x^4 - 30x^2 + 3) \\
L_5(u) &= \tfrac{1}{8}(63x^5 - 70x^3 + 15x) \\
L_6(u) &= \tfrac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5)
\end{aligned}$$

Assume that the function $f(x)$ is transformed by the scale change into the function $g(u)$. Least-square polynomial approximation of $g(u)$ is obtained by evaluating the coefficients b_k in sequence, using the defining equation

$$b_k = \int_{-1}^{+1} p_k(u) g(u)\, du \tag{21.6}$$

The process is terminated when the degree of the approximating polynomial is high enough to yield an acceptably close fit.

Note that Eq. (21.6) does not permit the use of a weighting function corresponding to $w(x)$ of Eq. (21.3). This is because the independence of the b_k on one another (and hence the elimination of the need to solve sets of linear equations) is based on the orthonormality of the $p_k(u)$, namely,

$$\int_{-1}^{+1} p_i(u) p_j(u)\, du = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$

If the inclusion of a weighting function is desired, a corresponding set of orthonormal polynomials must be found to yield an equation for the coefficients in the form of Eq. (21.6).

Piecewise continuous least-square polynomial curve fitting may be obtained by

applying Eqs. (21.5) and (21.6) to each segment of the approximation. The limiting abscissas of the various segments need not be specified but are preferably treated as additional parameters x_j which appear in $g(u)$ through Eq. (21.4). Each x_j corresponds to $u = +1$ for the approximating polynomial $P_{j-1}(u)$ and to $u = -1$ for the succeeding polynomial $P_j(u)$. The conditions

$$P_{j-1}(1) = P_j(-1) \qquad j = 1, 2, \ldots \tag{21.7}$$

are used to determine the x_j. These conditions not only assure continuity in the curve fitting but also guarantee minimum (or at least an extremum in) least-square error.

In the simulator the appropriate $P_j(u)$ or the equivalent $R_j(x)$ might be obtained by comparisons with the x_j in the manner indicated in Fig. 21.16. Computation of the polynomial is best mechanized by synthetic division, as shown in Fig. 21.17.

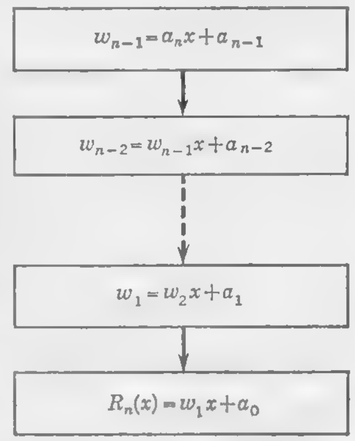


FIG. 21.17. Evaluation of an nth-degree polynomial.

Chebyshev Polynomial Approximation. This method seeks to minimize the maximum deviation of $f(x)$ from the approximating polynomial. Thus the error is generally larger in the center and smaller at the ends of the interval than for the corresponding least-square approximation. The method is based on the theorem that the maximum absolute value of $T_n(x)$, the Chebyshev polynomial of degree n, is smaller than the maximum absolute value of any other polynomial of degree n and leading coefficient unity.

The Chebyshev polynomials are defined over the interval $-1 \leq u \leq 1$ by the equations

$$\begin{aligned}
T_0(u) &= 1 \\
T_1(u) &= x \\
T_n(u) &= xT_{n-1}(u) - T_{n-2}(u)
\end{aligned} \tag{21.8}$$

which show that $T_n(u)$ is an nth-degree polynomial with unit leading coefficient. Equations (21.8) may be shown to be equivalent to

$$\begin{aligned}
T_0(u) &= 1 \\
T_n(u) &= 2^{1-n} \cos\,(n \cos^{-1} u)
\end{aligned} \tag{21.8a}$$

from which it follows that $T_n(u)$ oscillates between the extreme values $\pm 2^{1-n}$ and has n zeros over the interval from -1 to 1 given by

$$u_i = \cos \frac{(2i+1)\pi}{2n}; \; i = 0, 1, 2, \ldots, (n-1). \tag{21.9}$$

The most direct application of these equations is to choose as a Chebyshev approximation the nth-degree polynomial $P_n(u)$ which passes through the n points $g(u_i)$ where the u_i are the zeros given by Eq. (21.9). In this case the largest possible deviation $[P_n(u) - g(u)]$ cannot exceed $G^{(n+1)}/2^n \cdot (n+1)!$ where $G^{(n+1)}$ is the upper bound for the $(n+1)$th derivative of $g(u)$ over the interval.

A more effective but more time-consuming method of Chebyshev approximation is to expand $g(u)$ in a series of such polynomials. The procedure is somewhat simplified by introducing the modified Chebyshev polynomials and the new range

$$C_n(v) = 2 \cos\left(n \cos^{-1} \frac{v}{2}\right) \tag{21.10}$$

$$-2 \leq v \leq 2$$

Assume that the given function $f(x)$ transforms to $h(v)$. Then the approximation

$Q_n(v)$ is obtained from

$$Q_n(v) = \sum_{i=0}^{n} c_i C_i(v) \tag{21.11}$$

where

$$c_k = \frac{1}{2\pi} 2 \int_{-2}^{2} \frac{h(v) C_k(v)}{\sqrt{4 - v^2}}\, dv \tag{21.12}$$

Application of Eqs. (21.11) and (21.12) leads to the expansions

$$v^n = C_n(v) + \binom{n}{1} C_{n-2}(v) + \binom{n}{2} C_{n-4}(v) + \cdots \tag{21.13}$$

By expanding $h(v)$ in Taylor's series about $v = 0$ and substituting from Eq. (21.13) for the powers of v, another form is obtained for the c_k, namely,

$$c_k = \frac{h^{(k)}(0)}{k!} + \frac{h^{(k+2)}(0)}{1!\,(k+1)!} + \frac{h^{(k+4)}(0)}{2!\,(k+2)!} + \cdots \tag{21.14}$$

If for $k = n$ the first term on the right is the only significant term, $Q_n(v)$ should be a uniformly good approximation to $h(v)$ over the entire range.

If a polynomial approximation has been obtained by other means, say by least squares, the degree of the polynomial can sometimes be reduced by rearranging the approximation as a Chebyshev expansion $\sum_{i=0}^{n} c_i^* C_i(v)$ using Eq. (21.13). Since all $C_k(v)$ oscillate between ± 2, their contributions are in proportion to their coefficients c_k^*. If $|c_n^*| \ll |c_i^*|$, for all $i < n$, then the contribution of the nth-degree polynomial $C_n(v)$ can be ignored. Similarly for c_{n-1}, c_{n-2}, etc., until the first significant coefficient occurs.

Approximation by Ratio of Polynomials. In this method the function $f(x)$ is approximated by a ratio of polynomials $P(x)/Q(x)$. The method is particularly effective when $f(x)$ has one or more sharp peaks or troughs, as, for example, in airplane aerodynamic coefficients in the transsonic region. These rapid excursions imply large high-order derivatives and hence poor polynomial fitting as suggested by the remarks in connection with the simple method of Chebyshev approximation. The quadratic factors $\pm[a^2 \pm (x - b)^2]$ of $Q(x)$ are better able to fit peaks and troughs, particularly when $x_0 \le b \le x_s$ and $a \ll (x_s - x_0)$.

The method of least squares may be used by modifying D^2 of Eq. (21.3) as follows:

$$D^2 = \int_{x_0}^{x_s} w(x)\, [P_n(x) - Q_m(x) f(x)]^2\, dx \tag{21.14a}$$

$Q_m(x)$ behaves here like an initially unknown mth-degree polynomial weighting function. To correct for this weighting error, the curve fitting can be iterated as often as desired, dividing $w(x)$ at each step by $Q_m^2(x)$ from the previous step to obtain the appropriately modified weighting function.

Another approach is based upon the use of continued fractions. Let $f(x)$ be assumed to be expressed as the continued fraction

$$f(x) = a_0 + \cfrac{a_1}{1 + \cfrac{a_2}{1 + \cfrac{a_3}{1 + \cdots}}} \tag{21.15}$$

Rewrite this for convenience in the form

$$f(x) = a_0 + \frac{a_1}{1} + \frac{a_2}{1} + \frac{a_3}{1} + \cdots \tag{21.15a}$$

To determine the coefficients a_k, set

$$z = \phi(y) - x \tag{21.16}$$

where ϕ and f are inverse functions, i.e., $y = f(x + z)$. Choose $a_0 = y_0 = f(x_0)$ about midway between the extremes of $f(x)$. Note that this implies that $\phi(y_0) = x_0$ and $z_0 = 0$.

The next step is to determine the derivatives $z^{(r)} = d^r z/dy^r$ at $y = y_0$. If $f(x)$ can be inverted in closed form so that $\phi(y)$ is available, then $z^{(r)} = \phi^{(r)}$. However, if $f(x)$ exists only in graphical or tabular form, the derivatives must be computed from, say, divided differences. The relations

$$\frac{d}{dy} = \frac{dz}{dy}\frac{d}{dz} = \frac{1}{f'}\frac{d}{dz}$$

$$z'(y) = \frac{1}{f'}$$

$$z^{(2)}(y) = \frac{1}{f'}\frac{dz'}{dz} = -\frac{f^{(2)}}{(f')^3}$$

$$z^{(3)}(y) = \frac{3(f^{(2)})^2 - f' f^{(3)}}{(f')^5}$$

etc., can also be used up to the order needed for desired accuracy.

The derivatives are used to determine certain intermediate coefficients

$$p = -\frac{z}{z'} \qquad p_2 = \frac{1}{2}\frac{z^{(2)}}{z^{(1)}} \qquad p_k = \frac{1}{k}\frac{z^{(k)}}{z^{(k-1)}} \tag{21.17}$$

Note that x appears as a linear term in p and nowhere else. The a_k are given by

$$a_1 = p \qquad a_2 = pp_2 \qquad a_3 = p(p_2 - p_3)$$

$$(k+1)a_{k+1} = 2a_2 + (k-2)a_{k-1} - a_k - a_1 \frac{d}{dy} \ln \frac{a_k}{p} \qquad k \ge 2 \tag{21.18}$$

from which it follows that they are all linear functions of x. Note that $p = 0$ for $y = y_0$ and hence all $a_k = 0$ for $y = y_0$; thus the continued-fraction Eqs. (21.15) is exact for $f(x_0)$.

To show that the result is a ratio of polynomials, write $a_k = \alpha_k + \beta_k x$ and consider $f_m(x)$, the approximation to m terms, beginning with

$$\frac{a_{m-1}}{1 + a_m} = \frac{\alpha_{m-1} + \beta_{m-1} x}{(1 + \alpha_m) + \beta_m x}$$

It will be seen that the degree of the numerator is always equal to or one greater than that of the denominator.

If desired the residual error $[f(x) - f_m(x)]$ may be approximated as a polynomial to be added to the ratio $f_m(x)$.

21.4.4. Dynamic Behavior of Digital Simulators. The dynamic behavior of a simulated physical system is generally represented by a set of differential equations with time as the independent variable. Satisfactory simulation may call for the use of partial differential equations as, for example, in the case of heat flow through a conducting volume or vibrations of a flat plate. Whenever possible, however, the mathematical representation is approximated by a set of ordinary differential equations with the multidimensional dependence hidden in the now more complicated system coefficients. Only the ordinary case will be considered here.

Numerical Solution of Differential Equations. It is customary to specify the equations as expressions for the first derivatives of the dependent variables, thus:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, \ldots, x_m, t) \\ \dot{x}_2 &= f_2(x_1, \ldots, x_m, t) \\ &\cdots\cdots\cdots\cdots \\ \dot{x}_m &= f_m(x_1, \ldots, x_m, t) \end{aligned} \tag{21.19}$$

The functions f_k include the polynomial approximations or other curve-fitting functions discussed in the preceding section. The initial values of the dependent variables $x_k(t_0)$ are assumed given. Each set of initial values describes a specific initial state of the simulated system. The f_k include also certain time-dependent functions which represent the application of external forces. The solution of Eqs. (21.19) for a given initial state and specified forcing functions simulates a particular system response.

The numerical solution obtained by digital computation consists of a table of values of all m dependent variables at successive time instants $t_0, t_1, t_2, \ldots$, such that Eqs. (21.19) are obeyed to desired accuracy at every instant t_k. The interval between successive instants, $(\Delta t)_k = t_k - t_{k-1}$, need not be constant, but it is convenient and hence usual to use a constant $\Delta t = h$ except when the solution is being "started." The accuracy of the solution depends on the "quadrature formula" used to integrate Eqs. (21.19) and the size of the step h. Time of solution decreases as h is increased, but accuracy decreases as well. If the interval used is excessively large the variables x_k may even begin to oscillate in a spurious mode with unwarranted growth in magnitude and the solution may become unstable. It is important, therefore, to use a good quadrature formula with known measures for both accuracy and stability.

A simple method for starting the solution is to use the point-slope formula through which the new state at time t_{n+1} is located by means of the slope at the last known time t_n:

$$\begin{aligned} O_{11}\colon x_k(t_{n+1}) &= x_k(t_n) + hf_k(t_n) \\ h &= t_{n+1} - t_n \end{aligned} \tag{21.20}$$

The procedure is first to evaluate the f_k for all k, then to obtain the new values of x_k for all k. The label O_{11} indicates that (1) this is an open or predictor formula using only past values (up to t_n) and (2) it uses only one ordinate and one derivative.

The error introduced at each step of the solution is given by

$$l_{11} = \frac{1}{2} h^2 \frac{d^2x(\tau)}{dt^2} \tag{21.20a}$$

where $t_n \leq \tau \leq t_{n+1}$. In order to reduce the error to a safe level, the interval h must be kept very small. It should be noted that the error at each step incurs additional errors at succeeding steps; i.e., each error propagates through the solution. In fact, it can be shown that, unless h is kept very small, the O_{11} formula leads eventually to unstable solutions.

The digital-computer program for starting the solution is shown in Fig. 21.18, using O_{11} for only a few steps and a small interval h.

Any one of many known formulas can be used for extending the solution. A particularly interesting method suggested by W. E. Milne proceeds in two steps. Using the simplified notation $x_j = x_k(t_j)$ and $\dot{x}_j = f_k(t_j)$, these become

$$\begin{aligned} O_{13}\colon x'_{n+1} &= x_{n-3} + \tfrac{4}{3}h(2\dot{x}_{n-2} - \dot{x}_{n-1} + 2\dot{x}_n) \\ l'_{13} &= {}^{28}\!/_{90} h x^{(4)}(\tau') \end{aligned} \tag{21.21}$$

$$\begin{aligned} C_{13}\colon x''_{n+1} &= x_{n-1} + \tfrac{1}{3}h(\dot{x}_{n-1} + 4\dot{x}_n + \dot{x}'_{n+1}) \\ l_{13} &= -{}^{1}\!/_{90} h x^{(4)}(\tau) \end{aligned} \tag{21.22}$$

As before, O_{13} is an open formula yielding a first guess x'_{n+1} with error l'_{13}. This guess is used to obtain an approximate derivative $\dot{x}_{n+1}$ for use in the closed or corrector formula C_{13}. The latter produces the value x''_{n+1} with about 28 times reduction in error.

The fourth derivatives $x^{(4)}$ appearing in the equations for the errors are not necessarily equal; all that is known is that τ' lies in the range t_{n-3} to t_{n+1} and τ lies between t_{n-1} and t_{n+1}. However $x^{(4)}$ should change only slightly over such a short interval $(4h)$, from which it follows that an even more correct solution is

$$\begin{aligned} x_{n+1} &= x''_{n+1} - \frac{D}{29} \\ D &= x''_{n+1} - x'_{n+1} \end{aligned} \tag{21.23}$$

Equation (21.23) serves also to indicate magnitude of error at each step; D should be small and vary smoothly as the computation proceeds.

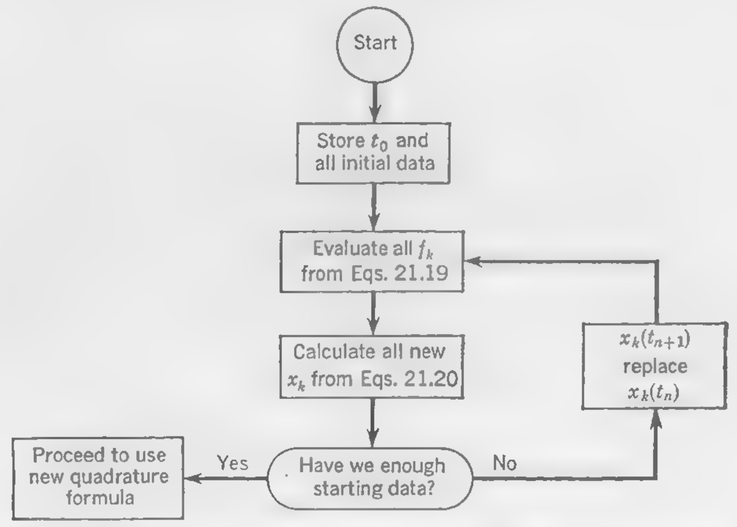

Start
Store t_0 and all initial data
Evaluate all f_k from Eqs. 21.19
Calculate all new x_k from Eqs. 21.20
Have we enough starting data?
Yes
No
Proceed to use new quadrature formula
$x_k(t_{n+1})$ replace $x_k(t_n)$

FIG. 21.18. Program to start the solution of a differential equation.

D serves also to monitor the permissible interval h. If $D \cong 0$, then h can be doubled, while if D becomes overly large, h should be halved. Simulation can thus be programmed to proceed at maximum speed consistent with acceptable accuracy.

Another popular procedure for solving ordinary differential equations is the Runge-Kutta method. There are many Runge-Kutta formulas, all characterized by their ability to compute each step of the solution without reference to past data. This is particularly attractive in that the need for a starting procedure is eliminated. Disadvantages of the Runge-Kutta methods are that they involve considerably more computation per step and the inherent error is difficult to estimate. The best procedure suggested to date for evaluating the accuracy of a Runge-Kutta simulation increases computing time about sixfold.

The most convenient Runge-Kutta formula for hand computation is the following:

$$x_k(t_{n+1}) = x_k(t_n) + \tfrac{1}{6}(q_{k1} + 2q_{k2} + 2q_{k3} + q_{k4}) \tag{21.24}$$

where
$$q_{k1} = hf_k(x_1, x_2, \ldots, x_m, t_n) \tag{21.24a}$$
$$q_{k2} = hf_k(x_1 + \tfrac{1}{2}q_{11}, x_2 + \tfrac{1}{2}q_{21}, \ldots, t_n + \tfrac{1}{2}h) \tag{21.24b}$$
$$q_{k3} = hf_k(x_1 + \tfrac{1}{2}q_{12}, x_2 + \tfrac{1}{2}q_{22}, \ldots, t_n + \tfrac{1}{2}h) \tag{21.24c}$$
$$q_{k4} = hf_k(x_1 + q_{13}, x_2 + q_{23}, \ldots, t_n + h) \tag{21.24d}$$
$$x_1 = x_1(t_n), \; x_2 = x_2(t_n), \ldots, x_m = x_m(t_n)$$

A modification which makes smaller demand on computer storage differs only in some of the coefficients, as follows:

(*a*) 1, $(2 - \sqrt{2})$, $2 + \sqrt{2}$, 1 replaces 1, 2, 2, 1 in Eq. (21.24).
(*b*) $x_i + \frac{1}{2}(\sqrt{2} - 1)q_{i1} + \frac{1}{2}(2 - \sqrt{2})q_{i2}$ replaces $x_i + \frac{1}{2}q_{i2}$ in Eq. (21.24*c*).
(*c*) $x_i - (1/\sqrt{2})\, q_{i2} + \frac{1}{2}(2 + \sqrt{2})q_{i3}$ replaces $x_i + q_{i3}$ in Eq. (21.24*d*).

In either case the procedure is to evaluate all q_{k1}, then all q_{k2}, q_{k3}, and q_{k4}. These terms are then substituted into the m equations for $x_k(t_{m+1})$ and the process is repeated. The error of either method is of the order h^5.

Real-time Solution of Differential Equations. In the preceding paragraphs it was assumed that the simulation was to be performed "off-line," as an independent computation which was acceptable in a slowed-down time scale. It is often required, as with simulators used for training personnel and with "on-line" computers used to control physical systems, that the simulation must be performed in "real time." In this case it is the responsibility of the digital computer to keep pace with the simulated process or device, so that the quadrature step h is a real time interval. The computer must therefore be fast enough to process in real time Δt all the arithmetic and decision-making entering into the quadrature step. Conversely a criterion must be provided for determining the maximum permissible quadrature interval h consistent with stability and reasonable accuracy for all possible simulator responses, and a quadrature procedure must be found for which $\Delta t \leq h$.

The criterion to be described here is based upon *a priori* knowledge (or computational determination) of the eigenvalues or natural small-amplitude frequencies of the simulated system in all its physically permissible or conceivable dynamic states. For each quadrature formula Q, the error introduced into the gross simulator response (not just at each step of the solution) is estimated from a "stability chart." The latter plots for all "true" frequencies λ_k the corresponding "false" frequencies μ_k which would appear in the numerical solution by method Q.

The typical stability chart shown in Fig. 21.19 is for the "nonclassical" formula "O_{33} mod Gurk"

$$x_3 = 1.1462084x_2 - 0.2010870x_1 + 0.0548787x_0 + 1.6415864\dot{x}_2 - 1.0080130\dot{x}_1 + 0.2750971\dot{x}_0 \quad (21.25)$$

Use of the formula requires that the starting conditions be given at t_0, t_1, and t_2 or be obtained in advance using O_{11} and the program of Fig. 21.18 or equivalent. In practice, satisfactory real-time simulation is obtained by assuming all derivatives zero and all ordinates constant initially. Simulation then proceeds by repeated application of Eq. (21.25) at fixed interval h.

The stability of the simulated response may be predicted from the stability chart as follows: Rewrite Eqs. (21.19) in the locally linearized form, obtained by Taylor expansion about the initial values $x_k(t_0)$ and dropping all terms in $[x_k(t) - x_k(t_0)]$ except the linear term. Equations (21.19) then take on the form of a set of linear differential equations with constant coefficients:

$$\dot{x}_i = \sum_{j=1}^{m} a_{ij}x_j + g_i(t) \qquad i = 1, 2, \ldots, m$$
$$a_{ij} = \frac{\partial f_i}{\partial x_j} \qquad \text{at } t = t_0 \quad (21.26)$$

The true solution of Eqs. (21.26) is

$$x_i(t) = \sum_{j=1}^{m} e^{\lambda_j t}\left[c_{ij} + \int_{t_0}^{t} e^{-\lambda_j \tau} z_i(\tau)\, d\tau\right] \quad (21.27)$$

where the c_{ij} are constants, each $z_k(t)$ is a weighted sum of the $g_i(t)$, and the λ_j are eigenvalues of the matrix (a_{ij}), i.e., the natural frequencies of the simulator in the initial state. Equations (21.27) show that the contribution of each natural frequency to the response of each variable x_k is an additive term independent of the other natural frequencies. Consequently system stability requires only that all λ_j have negative real parts.

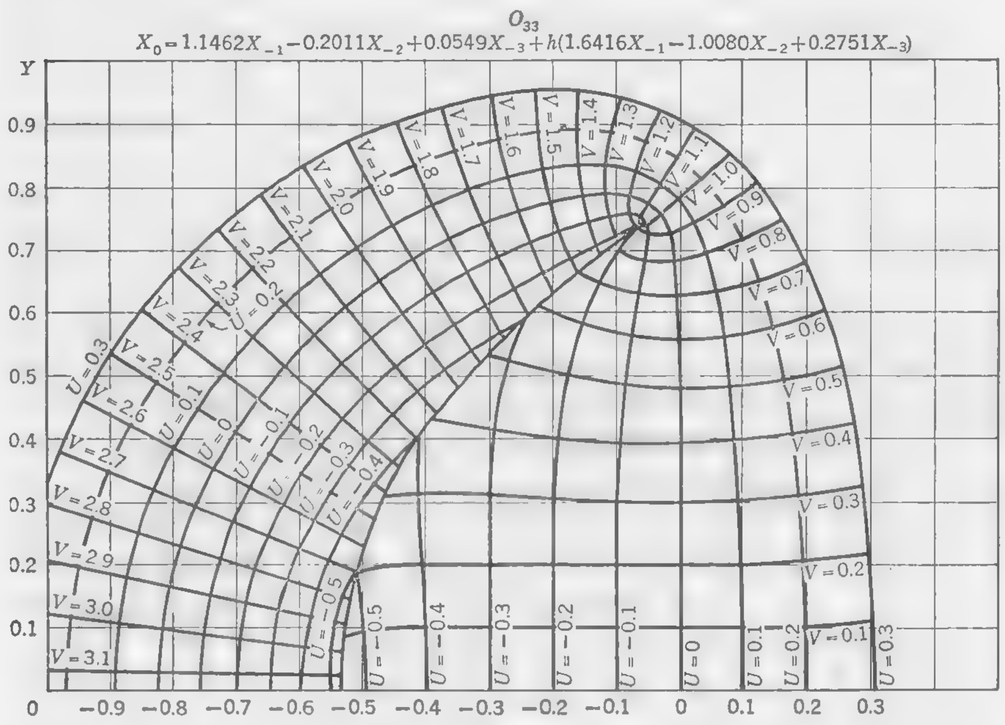


FIG. 21.19. Stability chart.

The behavior of the numerical solution of Eqs. (21.26) using quadrature Q may be determined similarly by examining the solution by Q of the following basic single equation with the single eigenvalue λ:

$$\dot{x} = \lambda x + g(t) \quad (21.28)$$

The numerical solution may be written in the form

$$x(nh) = \sum_{i=1}^{n} e^{\mu_i nh}\left(b_i + \sum_s G_{si}\, e^{-\mu_i sh}\right)$$
$$G_{si} = \frac{g_s}{\prod_{\substack{j=1 \\ j \neq i}} [1 - (r_j/r_i)]} \quad (21.29)$$
$$r_i = e^{\mu_i h}$$

where g_s is the contribution of the derivative terms in Q to the sth step and the b_i are determined by the initial conditions.

Note that the numerical solution contains N natural frequencies μ_k. If these are

enumerated in order of decreasing real part, then

$$|e^{\mu_1 h}| \triangleq |r_1| > |r_2| > \cdots > |r_n| \tag{21.30}$$

It follows from Eqs. (21.29) that stability requires $|r_1| < 1$. Again, comparing Eqs. (21.27) and (21.29), it follows that accuracy of simulation requires $\mu_1 \approx \lambda$ and $|r_1| \gg |r_2|$.

For the set of linear inhomogeneous Eqs. (21.26), the solution has the form

$$x_i(nh) = \sum_{k=1}^{m} \sum_{i=1}^{n} e^{\mu_k i n h} \left(b_{ki} + \sum_{s} G_{ksi} e^{-\mu_k i s h} \right)$$

where the index k refers as before to the eigenvalues λ_k of the matrix (a_{ij}).

The relation between the dominant frequency $\mu(\triangleq \mu_1)$ and λ using any quadrature formula Q can always be expressed as a mapping of the complex variable $w = \mu h = u + iv$ onto the complex plane $z = \lambda h = x + iy$, $i = \sqrt{-1}$. The result is the stability chart for Q. Note that the mapping is for the dimensionless frequencies w and z, taking the interval h into account; if all natural frequencies for all possible states of the simulated system are known in advance, the stability and accuracy of the simulation may be adjusted by choice of h and inspection of the appropriate stability chart.

For example, let the largest possible frequency of Eqs. (21.19) be $(-4 + 4i)$. Consider the numerical solution using O_{33} mod Gurk. The stability chart, shown in Fig. 21.19, is square out to about -0.25. Hence choice of $h = 0.05$ will yield accurate simulation since $w \cong z$ all the way out to maximum possible $z = -0.2 + 0.2i$. The choice $h = 0.1$ leads to slight inaccuracy since $w = -0.39 + 0.38i$ for $z = -0.4 + 0.4i$, but the simulation is stable. On the other hand, $h = 0.15$ leads to $w = (+0.07 + 2.19i)$ and hence an exponentially increasing, unstable response.

O_{33} mod Gurk is particularly suited to real-time simulation because (1) it requires relatively little computing per step and (2) its stability chart has a curve for $u = 0$ which hugs the y axis up to $y = 0.7$ and encloses a large area out to $x = -0.8$. Thus although accuracy becomes poor outside a small area in the neighborhood of the origin, stability is assured over a wide range of frequency using large intervals h. Note that, in real-time simulation, the digital computer must contain a clock to keep track of computing time and deliberately to quantize it to the full time h at each step.

The stability chart for a typical Runge-Kutta formula is essentially square. Both the accuracy and the stability are substantially better than for O_{33}, and over a far larger area. Moreover the Runge-Kutta formulas all yield only one frequency μ for each true frequency λ, given by

$$w = \ln (1 + z + z^2 + \cdots + z^M) \tag{21.31}$$

where M is the order of the method, i.e., the number of increments q_{kj} used. For example, $M = 4$ in Eq. (21.24). These formulas are therefore excellent for general simulation computations. On the other hand, they are poor for real-time simulation because they require excessive computation per step.

21.4.5. Mixed Digital-analog Simulation. The performance of an analog simulator can often be improved by associating with it a digital computer for function generation, filtering, nonlinear relationships, and/or decision making. Conversely a digital simulator which controls continuous motion such as instrument deflections or drive motors must of necessity be tied to analog devices. Such an operational digital simulator is essentially a sampled-data feedback control system or a digital servomechanism.

A basic component of such a simulator is the analog-to-digital converter or *sampler*, the output of which has the form of a train of narrow pulses. Let the output $x(t)$ of an analog device provide the input to a sampler, assumed zero for $t < 0$. The output

of the sampler, shown in Fig. 21.20, is given by

$$x^*(t) = \sum_{n=0}^{\infty} x(nT)\delta(t - nT) \tag{21.32}$$

where $\delta(\tau)$ is the unit impulse function,

$$\delta(\tau) = \begin{cases} 1 \text{ if } \tau = 0 \\ 0 \text{ if } \tau \neq 0 \end{cases} \tag{21.33}$$

Note that the sampling period T corresponds to the quadrature interval h of the previous section.

The Laplace transform of Eq. (21.32) yields

$$X^*(s) = \sum_{n=0}^{\infty} x(nT)\epsilon^{-nTs} \tag{21.34}$$

where $X^*(s)$ is by definition the Laplace transform of $x^*(t)$. It is convenient to introduce the new variable $z = \epsilon^{Ts}$, from which it follows that

$$X(z) = \sum_{n=0}^{\infty} x(nT)z^{-n} \tag{21.34a}$$

$X(z)$ is referred to as the Z transform of $x(t)$ sampled at the (implied) constant interval T.

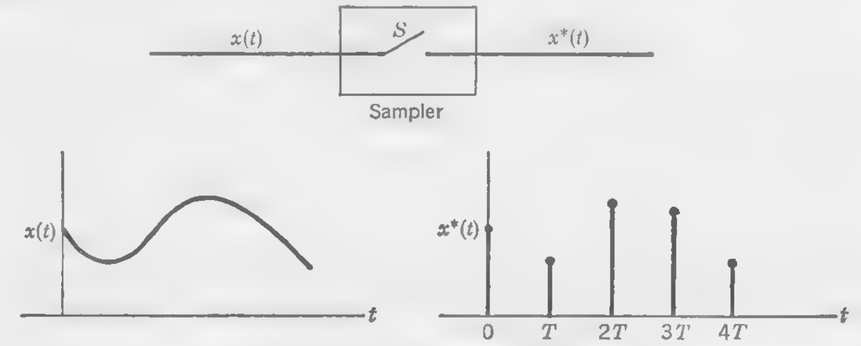


FIG. 21.20. Properties of a sampler.

If a sampled signal is applied as input to a network with impulse response function $h(t)$, the output of the network is given by

$$y(t) = \sum_{n=0}^{\infty} h(t - nT)x(nT) \tag{21.35}$$

The behavior of $y(t)$ at the sampling instants (even if y is not sampled) has the Z transform

$$Y(z) = H(z)X(z) \tag{21.36}$$

where both $Y(z)$ and $H(z)$ are defined from $y(t)$ and $h(t)$ by expressions of the form of Eq. (21.34a). Note the similarity to the expression for a continuous system for which the output $y(t)$ and the input $x(t)$ are related by the Laplace-transform relation

$$Y(s) = H(s)X(s) \tag{21.36a}$$

where $h(t)$ is the impulse response function of the system.

A primitive but typical operational digital simulator is illustrated in Fig. 21.21. Here, the input and output are sampled simultaneously and compared. The resulting digital difference $E_1(z)$ is used in the digital filter computation $D(z)$ to generate the sampled output error $E_2(z)$. The latter is applied to the analog filter $H(s)$ to generate the continuous output signal $y(t)$.

As a simple example of a digital filter, consider the case

$$D(z) = \frac{a + bz^{-1} + cz^{-2}}{p + qz^{-1}} \tag{21.37}$$

where a, b, c, p, q are constants. From Eq. (21.36) and Fig. 21.21

$$(p + qz^{-1})E_2(z) = (a + bz^{-1} + cz^{-2})E_1(z)$$

which, from the definition of z and the interpretation of ϵ^{-Ts}, corresponds to

$$pe_2(t) + qe_2(t - nT) = ae_1(t) + be_1(t - nT) + ce_1(t - 2nT) \tag{21.38}$$

Equation (21.38) expresses $e_2(t)$ as a function of the present value of $e_1(t)$, past values of both, and the constants of the digital filter. The filter may of course be made as intricate as desired.

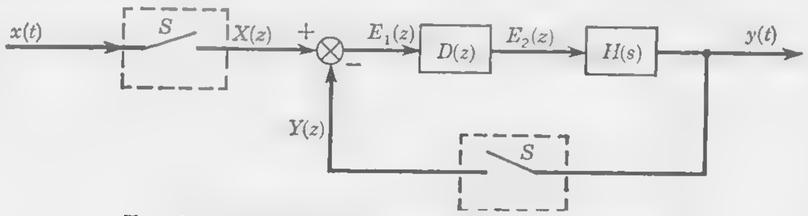


FIG. 21.21. A typical operational digital simulator.

The Z transform utilizes only the sampled data; the inverse Z transform similarly specifies only the behavior at sampled instants. The output between sampling instants can be obtained by assuming a doubled (tripled, etc.) sampling frequency and zero signal at the artificially introduced instants, as suggested by Fig. 21.20 and Eq. (21.32). The inverse of the revised Z transform then yields the output at the intermediate instants as well.

A feedback system such as that in Fig. 21.21 will be unstable if the Laplace transform of the system response has one or more poles to the right of the imaginary s axis. Since $z = e^{Ts}$, as s traces a contour consisting of the imaginary axis and a semicircle enclosing the right half plane, z traces a unit circle in the z plane. For stability, all poles must lie within the unit circle.

21.4.6. Examples of Digital Simulation. *Simulation of a Physical System.* A digital operational flight trainer, used to train pilots to fly new airplanes, is a typical example of simulation of a physical system, namely, the flight of an airplane. The particular airplane may be changed at moment's notice simply by replacing the stored data and instructions, and it is easy also to alter aerodynamic coefficients and/or flight conditions to test different models of the same airplane type traversing many flight paths. A single digital computer may simulate several airplanes simultaneously either in flight formation or in simulated combat.

Airplane response may be simulated both on the ground and in the air. Since rolling motion and certain airplane orientations are limited by ground forces, the ground equations necessarily differ from the air equations in some respects. The independent variable in either case is time: the primary dependent variables in flight are the linear velocities of the center of gravity u, v, w; the angular velocities of axes fixed in the airplane p, q, r; the direction cosines of these axes with respect to the vertical l_3, m_3, n_3; and the height above ground h. Other dependent variables of secondary nature or importance are angle of attack $\alpha = w/V$, where V is the airplane speed; Mach number $M = V/c$, where c is the local speed of sound; the direction cosines with respect to north and east; engine thrust and power-plant variables; airplane mass; and distance traversed north and east.

A typical program for a real-time operational flight trainer is shown in Fig. 21.22. The routine is entered at i when the start button is pressed. The entrance α leads to the first instruction "SIT h" which sets the interval timer to the real time h; thus h serves both as the time to complete each step of the flight solution and as the quadrature interval (see page 21–49). This is followed by inspection of the break-point switch "freeze"; if the simulation is temporarily suspended by the instructor while he briefs the trainee, the standby subroutine of Fig. 21.23 is entered through instruction β; if the flight is in progress, the program continues to the computations called for by box 1.

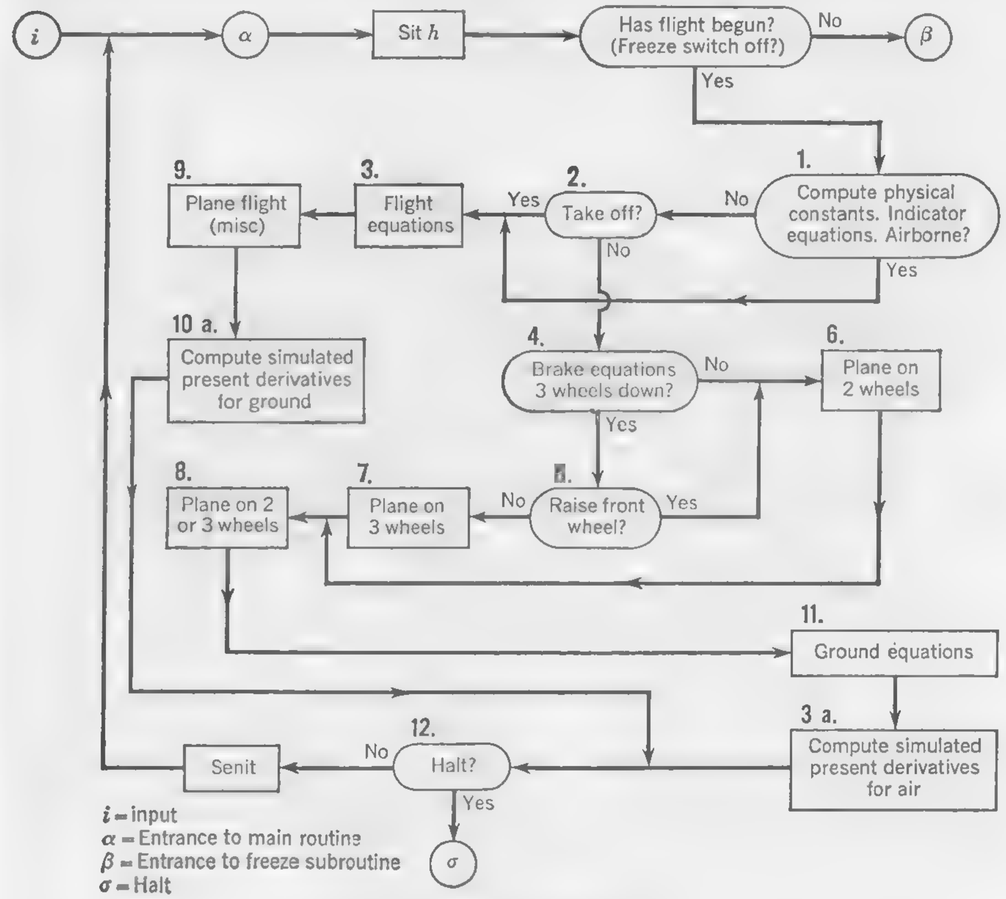


FIG. 21.22. Airplane flight simulation.

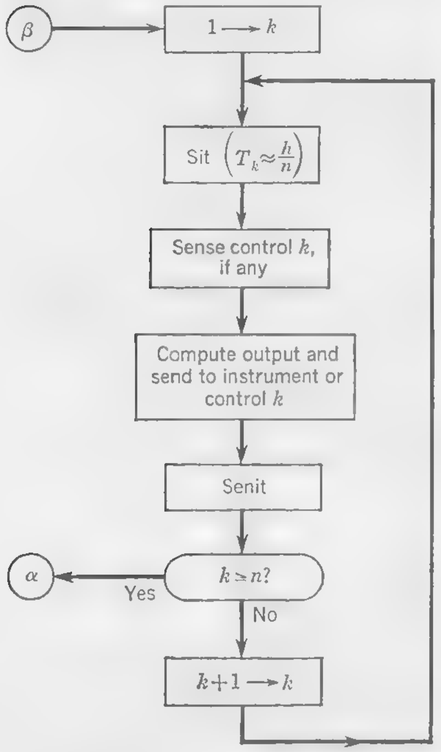
Box 1 refers to the program for computing data common to response on the ground and in the air. Some of the instrument data are computed here and relayed to the instrument panels in the cockpit and on the instructor's console. The box ends by asking if the airplane is in the air; if not, box 2 is entered to ask if the lift is sufficient to initiate take-off. The program has now determined whether box 3 or box 4 is to be entered.

Box 3 computes the next values of u, v, w, p, q, r, l_3, m_3, n_3, and h when in flight and leads to the program of box 9 where the secondary variables are computed. It turns out that maximum simulation speed is achieved by now computing ground data (box 10a) just in case the airplane lands during the next cycle of Fig. 21.22. The present cycle continues to box 12, where the simulation will be halted if the "halt" switch is operated on the instructor's console or the SENIT instruction will be called for. The latter senses the interval timer and causes the simulator to wait until the full interval h has passed, before permitting the next interval to begin.

Box 4 computes the effect of the brakes on ground speed. Boxes 4, 5, 6, and 7

then determine the angle between fuselage and ground. Secondary variables are computed in box 8 and the ground equations are integrated in box 11. Box 3*a* contains the program for computing variables for in-flight simulation in case the airplane takes off in the next interval. Box 12 and SENIT complete the cycle as before.

Note that the computations in boxes 3, 4, and 11 require sensing cockpit controls to determine control-surface deflections, throttle position, landing-gear position, and other pilot-induced parameters. An operational flight trainer is therefore a feedback control system with a human being in the feedback path. The signals fed back by the pilot are in response to stick-and-rudder forces (feel of the stick and rudder) and instrument readings.



β = Entrance to freeze subroutine
α = Entrance to main routine

FIG. 21.23. Freeze subroutine.

If flight is frozen, the subroutine of Fig. 21.23 is entered through β. The purpose of the freeze subroutine is to maintain the instrument readings and to lock the stick, rudder, and other controls. The SIT-SENIT instruction pairs space the outputs in accordance with the simulator requirements and the programmer's preferences. The total time to traverse the n cycles of the subroutine is preferably h, although it need not be. After each traverse the main program is entered through α to find out if the simulation is to be reinitiated. Note that the SIT h instruction is automatically obliterated by SIT T_1 when β is entered.

Locking the controls is easily programmed. The position P_0 of the control when the freeze was initiated is retained in the computer memory. The position P_t during freeze is sensed once per traverse of the freeze subroutine and a restoring force $F_t = K(P_0 - P_t)$ is applied to the control lever, where K is a large constant. If there is danger of oscillation or instability, the sampled-data theory discussed on page 21–54 may be used to derive a more satisfactory restoring force with a few terms based on previous positions; thus

$$F_t = K_1(P_0 - P_t) + K_2(P_0 - P_{t-1}) + \cdots$$

The components of a flight simulator are shown in the block diagram of Fig. 21.24. They may be described briefly as follows:

1. A library of digital programs and data for specific airplanes, stored on punched cards, punched tape, or magnetic tape
2. An input unit for loading the simulator memory with the program and data for a chosen airplane
3. One or more digital memory units for storing instructions, aerodynamic and engine data for the simulated airplane, and transient flight information
4. An arithmetic and logic unit for computing the aerodynamic forces, the flight equations one step at a time, the instrument readings, and the stick-and-rudder forces
5. Digital-to-analog decoder(s) for converting the digital information in the computer to analog voltages in order to actuate the instruments, generate stick and rudder forces, and provide continuous graphical records of selected flight variables
6. Analog-to-digital encoders for converting control-lever deflections into digital data for use in the digital computer

7. A cockpit for each airplane type to provide realistic environment for the pilot in the simulator, and a cockpit-selector switch

Simulation of Random Noise. The term "noise" refers to an unwanted signal which occurs simultaneously with and modulates, additively or otherwise, a wanted signal. Random noise refers to a signal for which the amplitude at any instant is completely independent of the amplitudes at all other instants. Random noise may be simulated in a digital computation by (an implicitly infinite but in fact finite) sequence of numbers with bounded range, say 0 to $(10^n - 1)$ and prescribed precision n, with a similar property of independence of each number from all the others in the sequence.

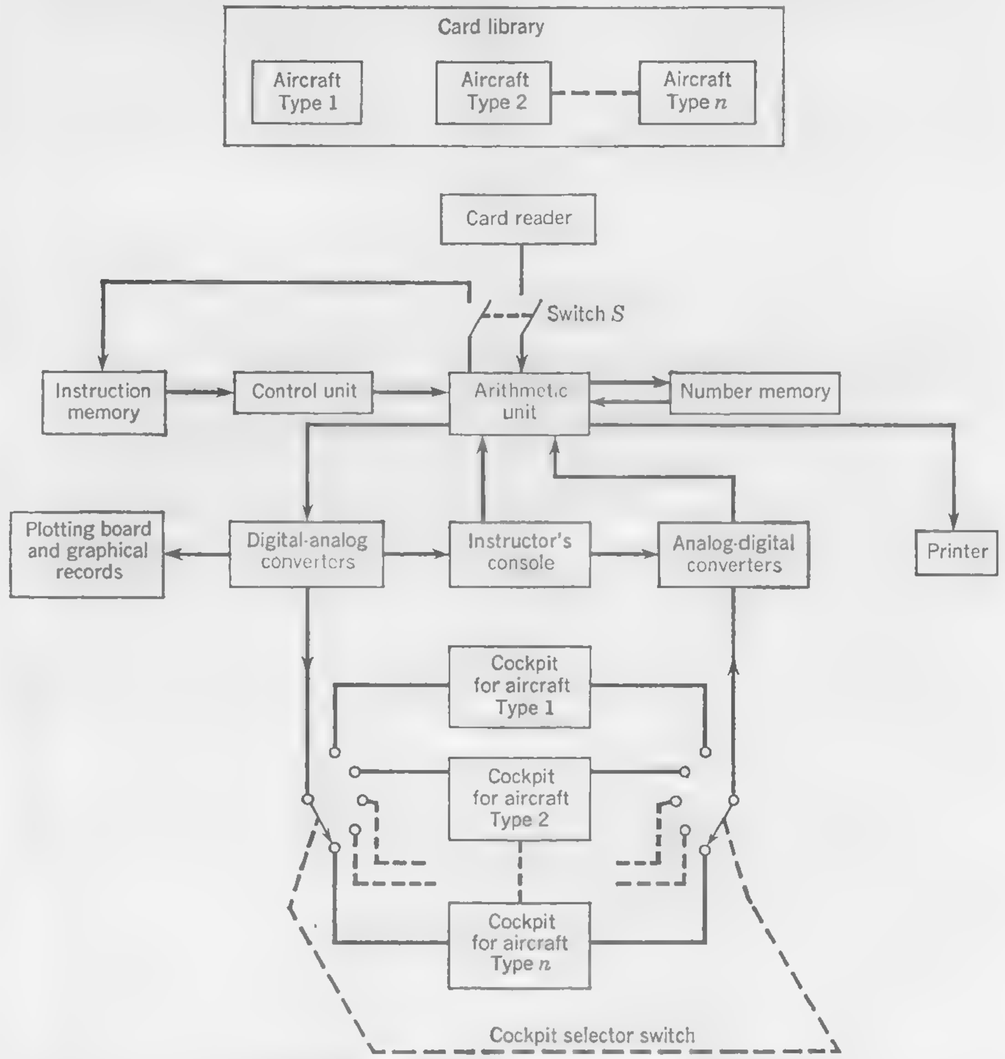


FIG. 21.24. Block diagram of a digital operational flight trainer.

Many tests have been devised for determining the approximation to true randomness of any given number sequence. A first test is merely to count the number of times each of the different numbers appears in the (finite) sequence. A second test is to count the number of times each different pair (triplet, etc.) of numbers occurs. A third test is the so-called poker test in which the digits of each number are compared on the basis of their value as a poker hand, i.e., one pair, two pairs, full house, etc. In each test the deviation of the counted distribution from the ideal random distribution may be used to determine the probability that the sequence is random; the probability serves in turn as a measure of confidence.

The digital computer can be used not only to generate random sequences but also to make the tests and to determine the grand total measures of confidence of each sequence. The simplest sequence comes from the Fibonacci numbers N_k obtained by the recurrence relation

$$N_k = N_{k-1} + N_{k-2} \tag{21.39}$$

where N_0 and N_1 are two unrelated n-digit numbers (say, relatively prime) with nonzero leading digits. By casting away carries into the $(n+1)$th column, which occurs automatically in a computer with n-digit precision, the residues of N_k modulo r^n are obtained. The number r is here the radix of the computer representation, 10 for decimal and 2 for binary. The residues generally form a long pseudo-random sequence which repeats only after a long cycle with length of the order of r^n. In a 35-bit binary computer this is equivalent to random noise which repeats only after more than a million samplings, which is certainly adequate.

Other recurrence relations for generating random numbers are

$$N_k \equiv cN_{k-1} (\text{mod } r^n) \tag{21.40}$$
$$N_k \equiv cN_{k-1} (\text{mod } r^n \pm 1) \tag{21.41}$$

where c is a prescribed constant, prime with respect to the modulus. These relations require multiplication, which is a slower operation than the addition and shifting of numbers in storage required by Eq. (21.39). Equation (21.41) suffers the additional disadvantage that the residue is not obtained automatically through computer overflow.

A typical example of the use of random noise arises in connection with the flight simulator described at the beginning of this article. The velocity of the airplane with respect to the air, needed for evaluating the aerodynamic forces, is the difference between the ground velocity of the airplane and the wind velocity. The latter may be represented by random "gusts" superimposed upon a steady average wind. The gusts can be computed along with the flight computations (box 3 of Fig. 21.22) with negligible effect on computation time.

Digital Simulation of Electric Networks. The voltage-current distribution in a large electric network can be computed digitally in a number of ways, one of which will be described here. The response of the network when any bus is short-circuited can also be simulated. The method is iterative, the final result being reached through successive approximations.

The following notation will be used:

N = number of nodes or junctions
V_i = potential at ith node
e_{ij} = net emf of generators in branch ij
g_{ij} = conductance of branch ij
$V_0, V_1, \ldots, V_{k-1}$ are the given (fixed) potentials
Superscript h refers to the hth iteration

$$I_{ij} = g_{ij}(V_i^h - V_j^h + e_{ij})$$

$$P_i^h = 2 \sum_{j=0}^{N-1} I_{ij} \qquad i = k, k+1, \ldots, N-1$$

$$S^h = \sum_{i=k}^{N-1} (P_i^h)^2$$

$$D^h = \sum_{i=k}^{N-1} \sum_{j=i+1}^{N-1} g_{ij}(P_i^h - P_j^h)^2$$

$$V_i^{h+1} = V_i^h - P_i^h \frac{S^h}{D^h} \qquad i = k, k+1, \ldots, N-1$$

The procedure is to iterate these equations until the first iteration which changes all the junction potentials by less than a specified small amount ϵ. The last values of V_i and I_{ij} are the junction potentials and branch currents being sought.

The computer flow chart is shown in Fig. 21.25. Short-circuited and grounded buses are simulated by appropriate assignments of the first k potentials V_i. The

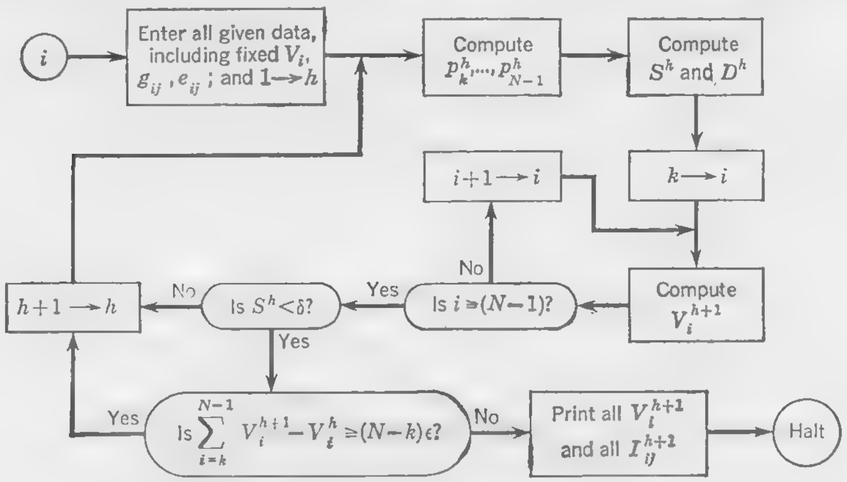


FIG. 21.25. Flow chart for network simulation.

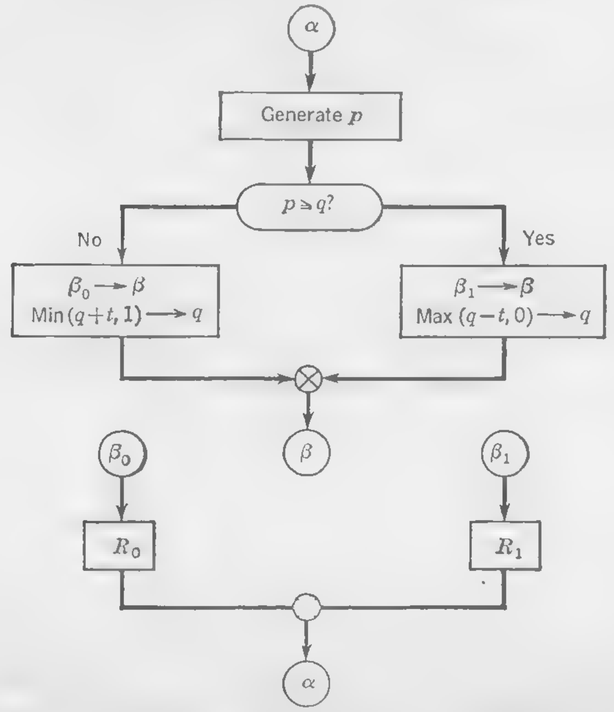


FIG. 21.26. Flow chart for simulation of habit forming.

arrow symbol is equivalent to the word "replaces." Note the use of a coarse test on S to accelerate the early iterations.

Simulation of Primitive Human Behavior. The references for this section discuss simulation of a wide variety of processes and devices, such as relay switching circuits, primitive biological evolution, traffic flow, telephone communication line loading, and

business management. A particularly interesting example is the simulation of human habit forming, which will now be described.

The discussion will be restricted to the simplest simulation of primitive habit forming. The flow chart for the digital-computer instruction program is shown in Fig. 21.26.

The "individual" is assumed to be born with a vacillating response to an environmental stimulus E. If E should be applied with strength greater than or equal to an inherited (or conditioned) inertia q, then the inertia will decrease by a small amount t and the individual will respond with behavior R_1. If E is applied with strength less than q, then q will increase by t and the behavioral response R_0 will result. The flow chart shows the effect of a single application of the stimulus E; the gradual habit forming results from repeated applications of E.

The assumption is made that the strength p of E is determined at each application by a random-number computation of p (as described on page 21–59). For simplicity, it is assumed that both p and q lie between 0 and 1, inclusive. The following can be proved:

1. The individual will gradually, but certainly, attain either an inertia $q = 0$ or $q = 1$. In either case it will develop a fixed unshakable nonvacillating response (R_0 or R_1, respectively) regardless of the strength of the stimulus E. In other words, it will form a habitual response.

2. The inherited inertia will bias the individual toward the "nearer" habit (e.g., R_0 if initially $q < \frac{1}{2}$) but the sequence of environmental stimuli will play an important role in the selection of the habit formed.

3. For uniformly distributed p, habit forming is rapid; it is almost certain that the individual will have formed a habit after about $1/t$ environmental stimuli, regardless of the initial value of q.

The simulation of human behavior can easily be rendered more complex by permitting multiple alternatives, by introducing many types of environmental stimuli, by inclusion of responses which react upon and control the environment, and so on. The limitations of the simulation model stem from our lack of understanding of the principles underlying human behavior and not from any limitation of the process of digital simulation itself.

21.5. COMPUTERS AND CONTROL*

Control equipment directing complex operations of a machine or a process often is required to perform some of the functions normally associated with computers. In a few instances, the control system may actually be identified as a full-fledged computer. But for the most part, the controls are so specialized and the computing problems so overshadowed by other tasks that they are not usually regarded as computers. The purpose of this section is to review some of these control systems, emphasizing in particular their computerlike aspects. The aim is not only to illustrate further applications of computers and computer technology but also, perhaps, to suggest new approaches to basic computer problems.

The systems presented in this section are limited to those in which the control equipment directly controls a machine or a physical process without human intervention, and in which the computing functions are of more than incidental significance. Thus data-logging techniques, as used in the process industries, and data-processing systems for business control, air-traffic control, and similar purposes are not discussed. Likewise, no attempt is made to treat conventional servomechanisms or regulators. Simulation techniques and simulators are not discussed, since they are amply covered elsewhere in the book.

The material of this section is organized somewhat arbitrarily into three major parts. Article 21.5.1, which follows this introduction, discusses primarily the control of machine tools. The same general techniques are applicable to many other devices which require precise mechanical positioning, possibly from programmed or stored data. Article 21.5.2 on equation-solving controls includes navigation and weapons-

* By E. Calvin Johnson.

control problems, as well as various process-control applications. In these controls, the continuous solution of algebraic and differential equations is usually of prime importance to the over-all operation. The section concludes with a description of digital-operational techniques and a brief discussion of their accuracy (Art. 21.5.3).

Another category of control systems, somewhat complementary to equation-solving controls, might be described as logical-decision controls. These systems derive their control action from sequences of logical steps or decisions, performed according to a prescribed set of rules. One example of this type is the control equipment for a telephone system, which through a series of logical operations establishes a communication channel between one telephone set and another. Recently developed versions of this equipment have much in common with conventional digital computers.[83] The modern traffic-controlled elevator, which is regulated automatically in accordance with the volume of traffic and the operation of adjacent elevators, is another example of a logical-decision system. Although controls of this general type appear to be growing rapidly in importance, further discussion is prevented by lack of space.

21.5.1. Machine-tool Controls. *Computing Functions of Machine-tool Controls.* A machine tool by definition is a device for shaping parts by the removal of material—usually in the form of chips—through relative motion between a cutter and the work. In many types of machines, the final shape of the workpiece is determined largely by the path followed by the cutter. In other types, the predominant factor is the shape of the cutter itself. Even in this case, the cutter must be properly located with respect to the work. Coordinated control of the various machine motions therefore is the principal job of a machine-tool control system. The performance of this job may require several computerlike operations.

In most cases, a part to be machined is described in an engineering drawing by a combination of lines and numbers. Lines convey forms and ideas; numbers add precision. Thus, in a large class of work, a necessary part of the original description is digital, even though the resulting part is fundamentally analog in nature. Control of the required machine motions therefore involves digital-to-analog conversion in one way or another. Conventionally the conversion is made by a human operator as he translates "blueprint" dimensions into machine motions. In numerically controlled machines, the conversion is a function of the machine-control system.

Automatic control of a machine through a complex cycle of operations requires some type of data storage. Cams and templates are simple means for storing data in analog form. The recording of continuous signals on magnetic tape is another analog method. Digital forms include punched cards, punched tape, and discrete signals recorded on magnetic tape. Data storage itself may be regarded as a computing function. In addition, the extraction of data from the storage medium may involve elementary computer operations of other types.

The amount of data which must be supplied to control the various machine motions may be materially reduced if the machine-control system is capable of generating elementary curves between specified points. This function is basically one of interpolation. It is of most significance as applied to machine operations such as milling and turning, in which the required cutter path may be rather intricate.

The functions of digital-to-analog conversion, data storage, and interpolation are representative of those which have been provided in existing machine-tool control systems (see Fig. 21.27). There are many other problems in the over-all process of converting a geometrical description into a finished part to which computer techniques may likewise be applied. These include determination of optimum cutting velocities; compensation for tool wear, servo dynamics, and other machine deficiencies; and compensation for cutter geometry where it is desirable to control the cutter center.

Some of the control techniques discussed in the following sections have found application to machines other than machine tools—textile machines,[23] inspection devices,[8] and automatic circuit-wiring machines,[127] for example. To date, however, machine-tool controls seem to have received the most attention from the standpoint of the number of problems attacked and the number of different approaches tried.

Digital-to-analog Conversion. The digital-to-analog conversion of principal concern in this section is that associated with the final machine motions. Other con-

versions of an instrument nature or of low accuracy frequently appear in machine-control systems, but their requirements do not differ appreciably from those of other common applications. The output conversion, however, does present some peculiar problems.

In most cases the analog quantity is linear motion of a machine slide, although it may also be rotary motion of a spindle. Frequently the power levels are fairly high and the load conditions, including adverse friction and cutting forces, widely varying. The least significant digits in the digital specification usually correspond to very small physical dimensions; 0.0001 in. is not uncommon. Required accuracies may

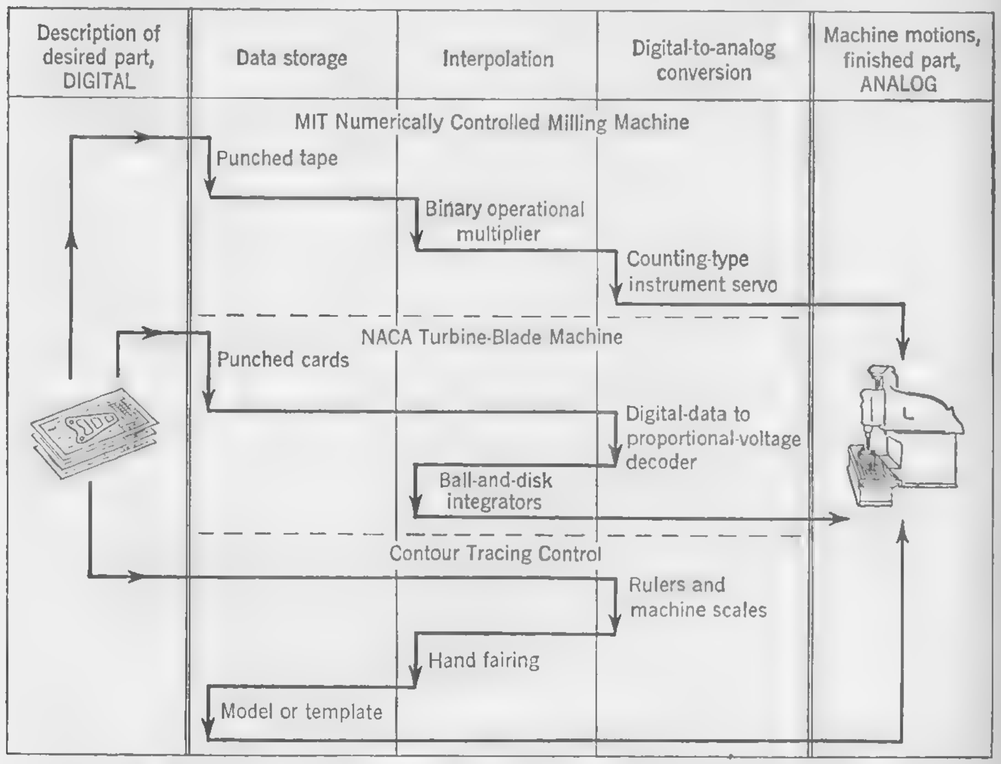


FIG. 21.27. Sequence of computing functions in some typical machine-tool systems.

well run to 1 part in 10^5—considerably better than normally considered usable in most analog forms.

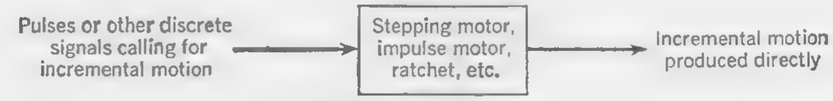
Machine-tool applications may be divided into two basic types according to their output requirements: (1) those in which end position only is important, as in locating the workpiece under a drill, and (2) those in which the position is important at all times, as in contour milling. In the latter category, positions must usually vary continuously. The method used in obtaining a digital-to-analog conversion depends upon which of these two types of outputs is involved and also upon the nature of the digital input information. Some control techniques most naturally lead to a set of discrete signals or pulses, each of which corresponds to a desired incremental movement. In other cases, the required position or change of position is represented by a number in binary or decimal code. These two classes of inputs are referred to as incremental and coded, respectively.

The conversion systems discussed in this section are grouped into four major categories: (1) those using stepping devices to produce incremental motions; (2) those using quantizers (e.g., pulse generators) to measure motion; (3) those using digitizers (e.g., coded commutators) to measure motion; and (4) those using matrix decoding schemes to generate essentially analog command signals. Figure 21.28 illustrates the functional distinctions among these four basic types. Principal

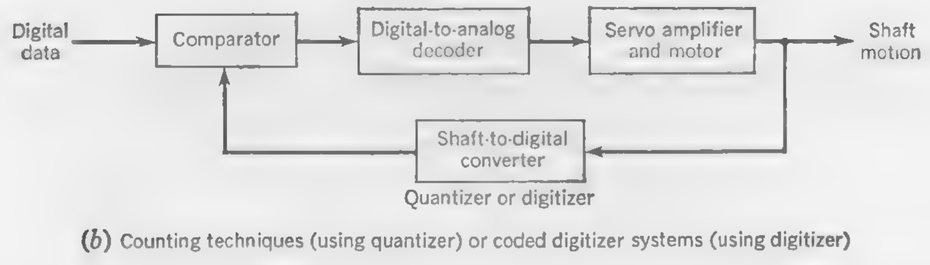
emphasis is placed on systems which directly convert digital information to machine motion. The same techniques, however, may be used to position an analog instrument such as a synchro. This then may provide analog command information to a conventional servomechanism that actually drives the machine member.

Stepping-type Devices. When the input is of the incremental type, a stepping motor serves conveniently as a simple form of digital-to-analog converter. Each discrete signal causes an incremental motion, which may be either geared to the moving machine member directly or coupled to it through a separate power servo. The machine member itself serves to integrate the individual increments.

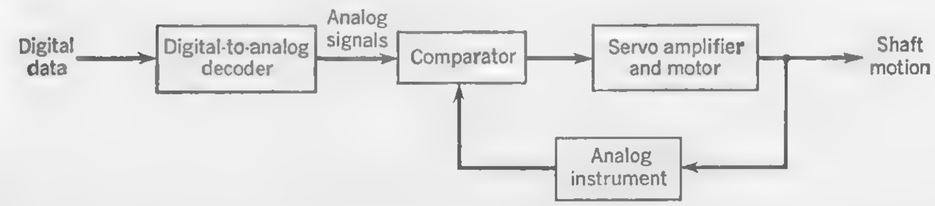
In most applications, the incremental input is either supplied by tapes and films of various types[29,141] or generated by the machine-control systems. In their simplest


Pulses or other discrete signals calling for incremental motion
Stepping motor, impulse motor, ratchet, etc.
Incremental motion produced directly

(*a*) Stepping-type devices



(*b*) Counting techniques (using quantizer) or coded digitizer systems (using digitizer)



(*c*) Conversion to analog commands

FIG. 21.28. Functional comparison of some basic digital-to-analog conversion methods. The schemes in (*b*) make use of precision analog-to-digital converters in a feedback loop to obtain an accurate digital-to-analog conversion.

forms, stepping devices are limited in speed and output torque. They are usually better adapted to continuous control of slowly varying motions than to end-point control where shifts must be made rapidly from one position to another.

The effect of a stepping device may also be secured through the use of two synchronous motors coupled by a differential in such a way that the speed of the output shaft is proportional to the difference between the individual motor speeds.[117] As indicated in Fig. 21.29, one motor is connected to a voltage of constant reference frequency, and the other to a voltage of controlled frequency. The output shaft runs at a speed and in the direction determined by the deviation of the control frequency from the reference. The output angle represents the integral of the difference between the number of cycles of alternating voltage supplied to the two motors. This system is able to provide high speeds and torques, but it is limited in dynamic response since changes in control frequency must be gradual enough that the motor does not slip or gain a pole.

Counting Techniques Using Quantizers. When stepping devices are not directly suitable because of speed, torque, or dynamic limitations, closed-loop principles may be applied.[91,149,71] Signals generated by a quantizer coupled to the moving member are compared with the incremental input signals. As shown in Fig. 21.30, for example, a reversible binary counter may be used to count continuously the difference between the number of input increments and the number of feedback increments. The net count standing in the counter at any time represents the position error in the servo loop. This count is converted to a proportional voltage which is then amplified to drive a servo motor.

This system can handle practically unlimited input rates with any type of motor having adequate dynamic performance and power capability to drive the load. If

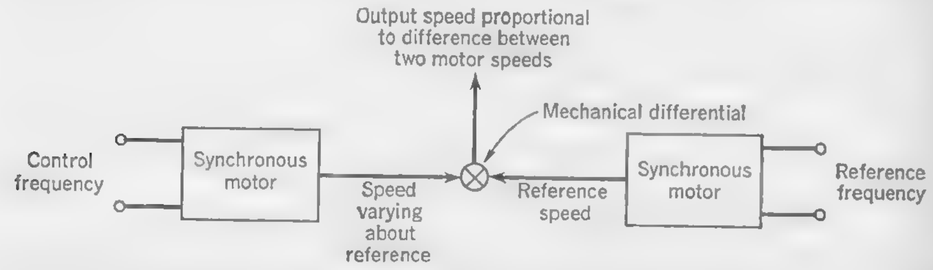


FIG. 21.29. Connection of two synchronous motors to provide the effect of a high-speed stepping motor. Under suitable restrictions, the output shaft angle represents the difference between the number of cycles of alternating voltage applied to the two motors.

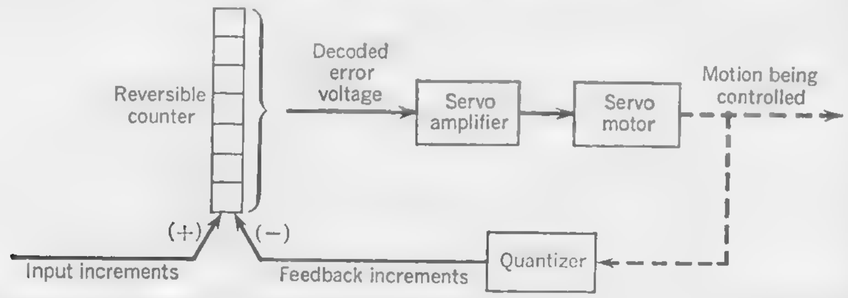


FIG. 21.30. Closed-loop incremental system using a quantizer as a feedback instrument.

the possible counting rates are sufficiently high and the quantized increments sufficiently small, little deterioration in servo performance results from the discrete nature of the feedback and error-sensing devices in the loop. The count-to-voltage conversion which is required takes place within the servo loop, where accuracy is not important. A good null is desirable, but decoding of only the first few stages in the counter usually provides adequate linear range for normal operation. The principal disadvantage of this system and all others employing quantizers is that accurate knowledge of position depends on having correctly counted all increments from the last known reference point. However, additional means to avoid difficulty on this account can frequently be justified by the relative simplicity of quantizers and their associated circuitry as compared with absolute digitizers, particularly in the case of control systems which naturally produce incremental data.

An approach related to the above is possible when the input is in coded form and only end points are being controlled digitally, with speed regulated in some other way if necessary.[102] In this case, a counter having sufficient stages to represent the longest desired stroke is preset by the input number. The drive motor then operates until the quantizer pulses the counter down to zero. The drive may be of the servo type if the counter is provided with an error-decoding matrix, or it may be simply a conventional motor from which power is removed when the count reaches zero. In an intermediate category, gear ratios between drive motor and machine may be changed

one or more times to slow the machine down at predetermined distances from the end point. An automatic reversing sequence is possible also, causing the final approach to be made always from the same direction and with the same speed in order to increase reproducibility and reduce inaccuracies due to backlash.

The types of quantizers used in machine-tool controls include many of those found in other conversion applications. Considerable emphasis, of course, is placed on reliability and long life in a relatively unfavorable environment from the standpoint of dirt, oil, and vibration. As a result, sliding electrical contacts as in slip rings and commutators are usually avoided when at all feasible. Also, high resolution is desirable in rotary devices in order to minimize the gearing required for coupling to a lead-screw or rack-and-pinion drive. A magnetic-induction quantizer providing 1,000 counts per revolution (with indication of direction of rotation) is typical of those capable of meeting both reliability and resolution requirements.[150]

For extremely accurate control of linear motions, a linear quantizer mounted directly to the machine member is desirable. Undue requirements on the accuracy and

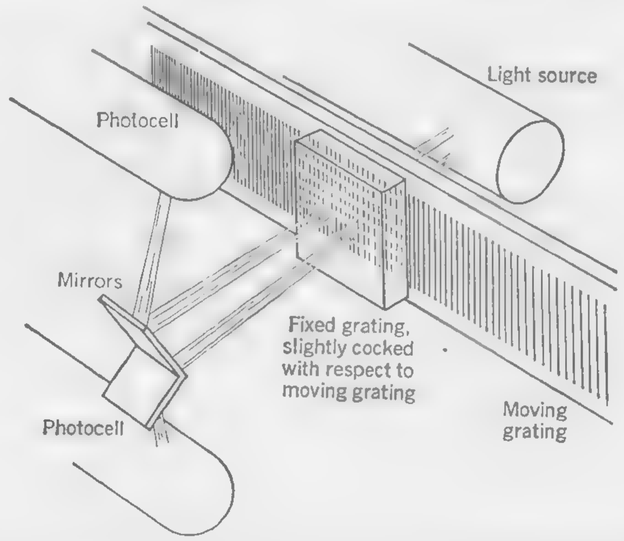


FIG. 21.31. Linear quantizer for direct mounting on a machine-tool slide. Optical gratings are used for high resolution. (Adapted from Fig. 2, Ref. 35.)

rigidity of lead screws and gearing can thereby be avoided. One type of high-resolution linear quantizer utilizes an accurately ruled optical grating the same length as the stroke to be measured (Fig. 21.31). Another short grating of the same pitch serves as a cursor. This latter grating is cocked slightly so that, when illuminated from behind, light and dark bands appear to move across the gratings in a Moiré pattern, perpendicular to the relative motion of the gratings. By observation of the bands photoelectrically at two different phases, information is available from which the direction of motion can be determined. A system of this type was built with gratings having 5,000 lines per inch to provide a resolution of 10,000 counts per inch.[35] Another system constructed for experimental purposes used two gratings having their rulings oriented parallel to produce a picket-fence effect. The image of the main long grating was magnified ten times optically and observed through a coarser secondary grating of the same effective pitch.[18]

Coded-digitizer Systems. The coded-disk type of digitizer is applicable to machine-tool controls with much the same techniques as are common elsewhere. In continuous-control applications, the reading of the digitizer is compared with a number representing the desired position, and any discrepancy is used to correct the actual position. If a suitable computer is available as part of the control system, the comparison may be made periodically as a conventional subtraction.[114] Otherwise, some

type of comparing circuitry may be employed which develops a signal proportional to the error, perhaps over only a limited range.

In end-point controls, a simpler comparison is adequate which detects coincidence only, or at most a limited number of points on the approach. Any of the drive schemes mentioned in the preceding section are applicable.[28,108,11]

Conversion to Analog Commands. A common problem is that of developing a position in accordance with the static representation of a dimension, as from a keyboard. In such a case, the keyboard can operate a precision voltage divider to produce a voltage proportional to the dimension (Fig. 21.32). The feed drive is actuated

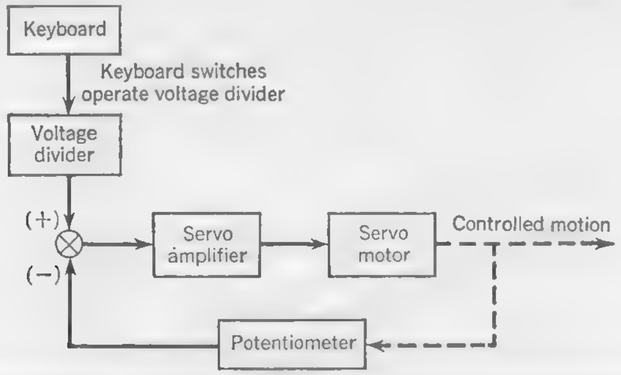


FIG. 21.32. Digital-to-analog conversion by keyboard-operated precision voltage divider.

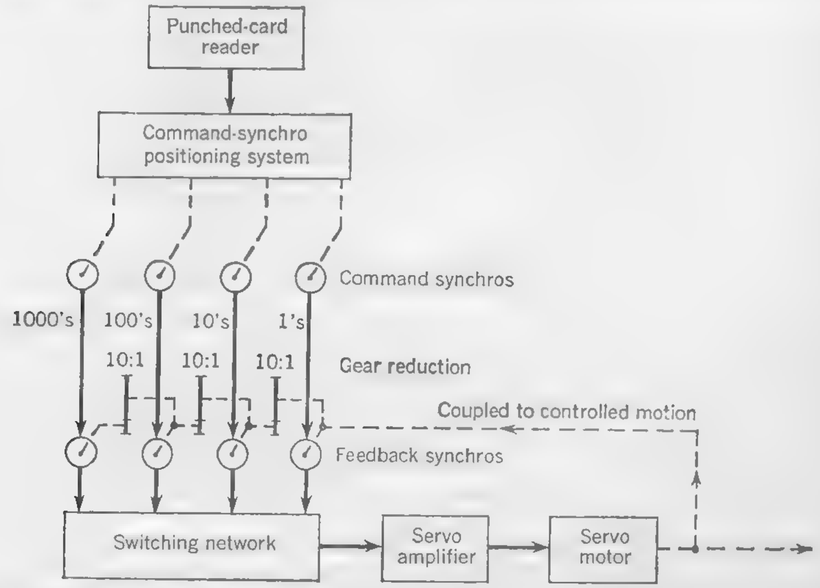


FIG. 21.33. Multispeed synchro applied to a machine-tool control. One pair of synchros is included for each decimal digit.

until a potentiometer coupled to it balances the keyboard voltage, indicating that the correct machine position has been assumed. By appropriate design of the voltage divider, command voltages may be developed similarly for synchros or resolvers. The keyboard in effect simulates a synchro transmitter or control transformer. Punched-tape or punched-card inputs can be handled in the same way, although intermediate relay storage may be necessary.[29,13]

If greater accuracy is required than can be obtained easily with a single analog instrument, multispeed data-transmission systems may be used. Figure 21.33 shows an example in which a separate pair of command and feedback synchros is provided

for each decimal digit. A commercial version, having a punched-card input, contains four synchro pairs.[75,129] The command synchros are positioned to the correct tenth of a revolution by stops linked mechanically to the card-reading pins. Further corrections are made by an arrangement of interconnecting planetary or differential gears, so that each synchro reflects the position of all those corresponding to digits of less significance. The most significant synchro, for instance, represents the entire command number, although not to the accuracy required. Feedback synchros are geared to the machine motion at ratios differing by factors of 10. The slowest feedback synchro is first compared with the most significant command synchro, and a drive motor is operated until a near null is obtained. Control is then switched to the next pair of synchros, which position within the range of the succeeding pair. This process is repeated until a balance is reached by the highest-speed synchro pair.

Interpolation. The role of interpolation in machine-tool controls is not unlike that in most other applications, being principally the generation of a smooth surface between a limited number of discrete points. In this instance, the discrete points define the contour to be followed by a cutter in order to produce the desired part. If the contour is truly three-dimensional, one machine motion may be considered to be a function of two others, in which case bivariate interpolation is appropriate. In most machine-control systems developed to date, however, the three-dimensional effect, if required, is obtained in two steps by successive applications of interpolating functions of one variable. These functions may take many forms; those which have been of most use so far are of the polynomial type:

$$y = f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \tag{21.42}$$

The coefficients a_k are usually determined by requiring that the nth-degree polynomial pass through $n + 1$ given points. In some cases it may be desirable to replace some of the $n + 1$ points by other conditions which might, for example, require continuity of slope between the polynomials for successive sets of points. As yet, little advantage appears to have been taken of such possibilities in machine-control applications.

The concept that one machine motion is a function of another (or that one motion is a function of two others) is useful for the cam and turbine-blade machines to be described, and for other machines such as planers in which one machine motion is always in the same direction and the rate of change of the others is limited. However, these conditions do not apply to all classes of work.

Consider the problem of milling an irregular closed contour in the xy plane, for example. Over portions of the contour where the variation of y with respect to x is relatively mild, y may be regarded as a function of x. However, for other portions of the contour, y is more appropriately regarded as the independent variable and x the dependent variable. Otherwise double-valued functions are required. Unless coordinates are shifted from time to time to avoid these difficulties, simple polynomial interpolation as indicated by Eq. (21.42) is not suitable. A more general form

$$F(x,y) = 0 \tag{21.43}$$

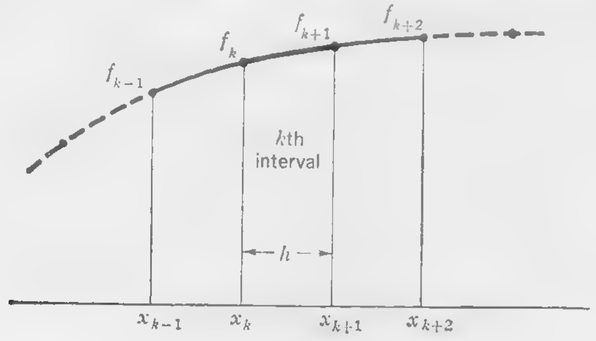
is desirable.

Linear Analog Interpolation. The effect of a linear, or first-degree-polynomial, interpolation with respect to time is obtained by simply maintaining uniform velocity between two points. If the same time base is used for two or more motions, the interpolation is linear with respect to each motion. For example, a taper can be cut on a lathe by regulating the velocities of the cross slide and the longitudinal carriage appropriately. The cross-slide velocity should be such as to cause the distance required along that axis to be covered in the same time needed by the carriage to complete its motion.

The accuracy of a taper cut in this manner depends upon the accuracy with which the velocities are regulated and upon the accuracy with which the end points are established. For short distances, conventional velocity-servo techniques, utilizing precision tachometers for feedback, can provide good accuracy. In one instance, a numerically controlled lathe has been reported to be capable of cutting tapers to

within 0.1 deg. This machine is programmed by a punched tape which supplies end points and feed rates in coded form. Precision voltage dividers generate voltage commands which are balanced by potentiometers in the case of end points and by tachometers in the case of feed rates.[29,102]

Cubic Analog Interpolation. Higher-degree interpolation can be achieved by well-known analog-computer techniques. An example is the cubic interpolator developed at the NACA Lewis Flight Propulsion Laboratory for control of a turbine-blade milling machine.[96,95] This device uses three ball-and-disk integrators to generate a third-degree polynomial which passes through four evenly spaced data points, as



x_{k-2}	f_{k-2}				
		$\Delta'_{k-1.5}$			
x_{k-1}	f_{k-1}		Δ''_{k-1}		
		$\Delta'_{k-0.5}$		$\Delta'''_{k-0.5}$	
x_k	f_k		Δ''_k		$\Delta^{(4)}_k$
		$\Delta'_{k+0.5}$		$\Delta'''_{k+0.5}$	
x_{k+1}	f_{k+1}		Δ''_{k+1}		
		$\Delta'_{k+1.5}$			
x_{k+2}	f_{k+2}				

FIG. 21.34. Third-degree polynomial set up for interpolating within the kth interval. The polynomial passes through the four surrounding data points.

indicated in Fig. 21.34. The section between the two central points is used for active control; as the machine moves on to the next interval, the polynomial is redetermined to include the next data point and omit the most remote.

The interpolating formula employed over the kth interval is

$$f(x) = f_k + \Delta'_{k+0.5}\theta + \tfrac{1}{2}\Delta''_k\theta(\theta - 1) + \tfrac{1}{6}\Delta'''_{k+0.5}\theta(\theta + 1)(\theta - 1) \qquad (21.44)$$

or

$$f(x) = f_k + (\Delta'_{k+0.5} - \tfrac{1}{2}\Delta''_k - \tfrac{1}{6}\Delta'''_{k+0.5})\theta + \tfrac{1}{2}\Delta''_k\theta^2 + \tfrac{1}{6}\Delta'''_{k+0.5}\theta^3 \qquad (21.45)$$

in which $\Delta'_{k+0.5} = f_{k+1} - f_k$ and $\Delta_j^{(m)} = \Delta_{j+0.5}^{(m-1)} - \Delta_{j-0.5}^{(m-1)}$. The independent variable x is equal to $x_k + \theta h$ where h is the interpolation interval and θ varies from 0 to 1 across the interval. The first, second, and third derivatives of $f(x)$ are

$$\frac{df}{d\theta} = (\Delta'_{k+0.5} - \tfrac{1}{2}\Delta''_k - \tfrac{1}{6}\Delta'''_{k+0.5}) + \Delta''_k\theta + \tfrac{1}{2}\Delta'''_{k+0.5}\theta^2 \qquad (21.46)$$

$$\frac{d^2f}{d\theta^2} = \Delta''_k + \Delta'''_{k+0.5}\theta \qquad (21.47)$$

$$\frac{d^3f}{d\theta^3} = \Delta'''_{k+0.5} \qquad (21.48)$$

The required polynomial $f(x_k + \theta h)$ can be produced by three successive integrations with respect to θ, starting with $\Delta'''_{k+0.5}$. Figure 21.35 shows three integrators arranged to accomplish this.

If the integrators are correctly set initially, it may be shown that a simple adjustment of the integrands of the first and third integrators is sufficient in proceeding from one interval to the next. Immediately before the kth interval, $\Delta_k^{(4)}$ should be added to the first integrand and $-\frac{1}{6}\Delta_k^{(4)}$ to the third (assuming θ starts over at 0 for each interval). Ideally, then, only the fourth difference associated with each interval need be supplied to the interpolator. In practice it was found desirable to supply the actual data points, as well, in order to check the computed values at the ends of the intervals. Any discrepancies detected were corrected over the next interval by an adjustment of the third integrand, thereby preventing the gradual accumulation of errors due to inaccuracies in the computing elements.

In another version of the cubic interpolator, one of the three integrations is obtained by means of a precision potentiometer.[104] The potentiometer bridges across two voltages, preset in accordance with appropriate differences, and its shaft is rotated with the independent variable. The potentiometer in effect then interpolates linearly

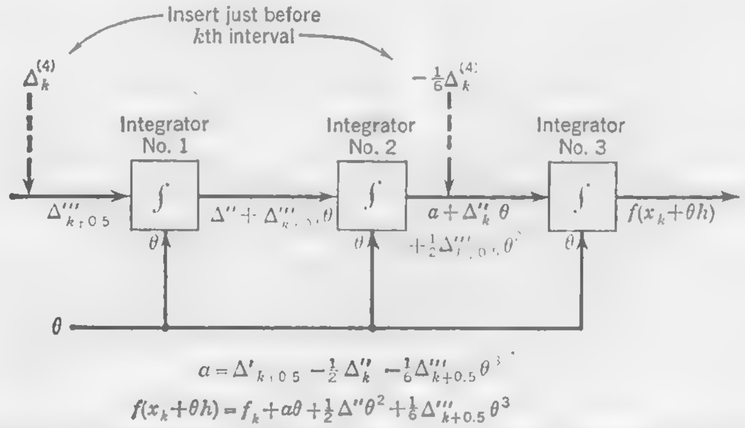


FIG. 21.35. Interconnection of integrators to produce third-degree polynomial in θ.

between the two voltages. In this latter system, the nature of the input data is somewhat different, but in both systems information is supplied by means of punched cards, one card per interval. An experimental model of the second system is reportedly able to produce a voltage corresponding to Eq. (21.45), having a maximum deviation from it of 0.15 per cent. The over-all system is accurate to within 0.1 per cent of the total excursion of the dependent variable.

Spline Interpolation. The curve assumed by a flexible metal strip or spline, constrained to pass through a small number of specified data points, is the basis of another NACA interpolator.[95] The complete system, illustrated in Fig. 21.36, consists of two identical splines. Each spline is positioned by knife-edges at four equally spaced points along its length in accordance with specified values of the dependent variable. These values are read from punched cards as four 3-decimal-digit numbers which are converted to voltages by means of precision voltage-divider networks. Each knife-edge is driven by a servo motor and lead screw until an associated potentiometer balances the divider voltage.

Only the center intervals of the splines are used for control purposes. These sections are traced by a probe connected to a high-voltage source to cause continuous arcing to the spline. A fixed gap between probe and spline is maintained by servoing the spline in accordance with the voltage drop across the arc.

Two splines are included in the system to permit continuous operation. One spline is traced while the other is being positioned. In Fig. 21.36, for example, the probe is driven up the left spline by a cam as the knife-edges supporting the right spline are set up for the next interval. When the boundary is reached, the probe is shifted to the right spline and moved downward in synchronism with the independent variable.

Digital Interpolation. The interpolation function can also be performed on a discrete or digital basis, although experience reported to date has been limited to linear types. A classic example is the numerically controlled milling-machine system developed at MIT (Fig. 21.37).[90] The metal-cutting element of the system is an adaptation of a conventional vertical-spindle milling machine, providing tool motion along three mutually perpendicular axes. Operation of the machine is governed by a punched tape which continuously supplies blocks of elementary instructions to the machine-control unit, or director. Each instruction block specifies: (1) the incremental motion required along each axis and (2) the time interval in which all motions are to be executed. The nature of the control system is such as to spread all motions evenly over the allotted time interval. The result is a straight-line tool path between two points in space—or, in effect, a linear interpolation.

The heart of the control system is a device called the pulse distributor, which more generally may be referred to as a binary operational multiplier. Its function is to

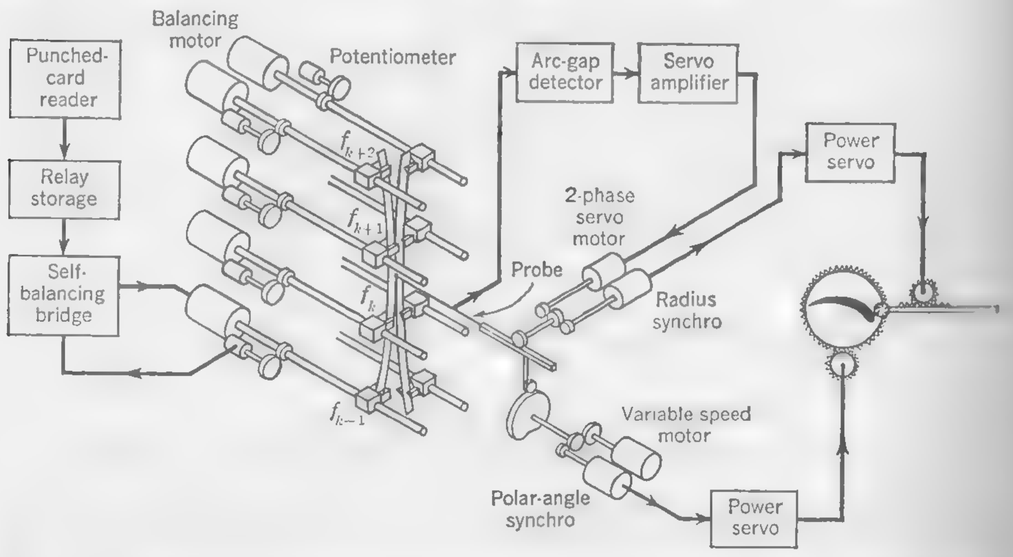


FIG. 21.36. Spline interpolator applied to a turbine-blade milling machine. (Adapted from Fig. 9, Ref. 94.)

produce three pulse trains—one for each machine axis—containing a number of pulses proportional to the motion required by a given instruction. Each pulse in the MIT system calls for an incremental motion of 0.0005 in. Such pulses are translated to rotary motion at an instrument level by counting-type servos. These servos, in turn, position synchros which supply analog command information to the actual machine-drive servos.

The pulse distributor, which is responsible for the interpolating ability of the machine, consists basically of a 10-stage binary counter (Fig. 21.38). Each stage in the counter is considered to produce two groups of pulses, called "carry" pulses and "noncarry" pulses. The carry pulses, which occur when the flip-flop changes from the 1 state to the 0 state, are passed along to trigger the next stage in the counter. The interspersed noncarry pulses, representing transitions from 0 to 1, are segregated as potential command pulses.

Since each successive stage of the counter divides the input pulse rate by 2, the tenth stage produces a carry pulse for every $2^{10} = 1{,}024$ input pulses. If the input rate is 512 pulses per second, such an end-carry pulse is produced every 2 sec. During each 2-sec interval, the first stage emits 512 noncarry pulses, the second stage 256, and so on down to the tenth stage, which emits 1. Any number of pulses Δx (up to 1,023) may be obtained during a 2-sec interval by gathering noncarry pulses from appropriate stages. As may be demonstrated, pulses should be taken from every stage

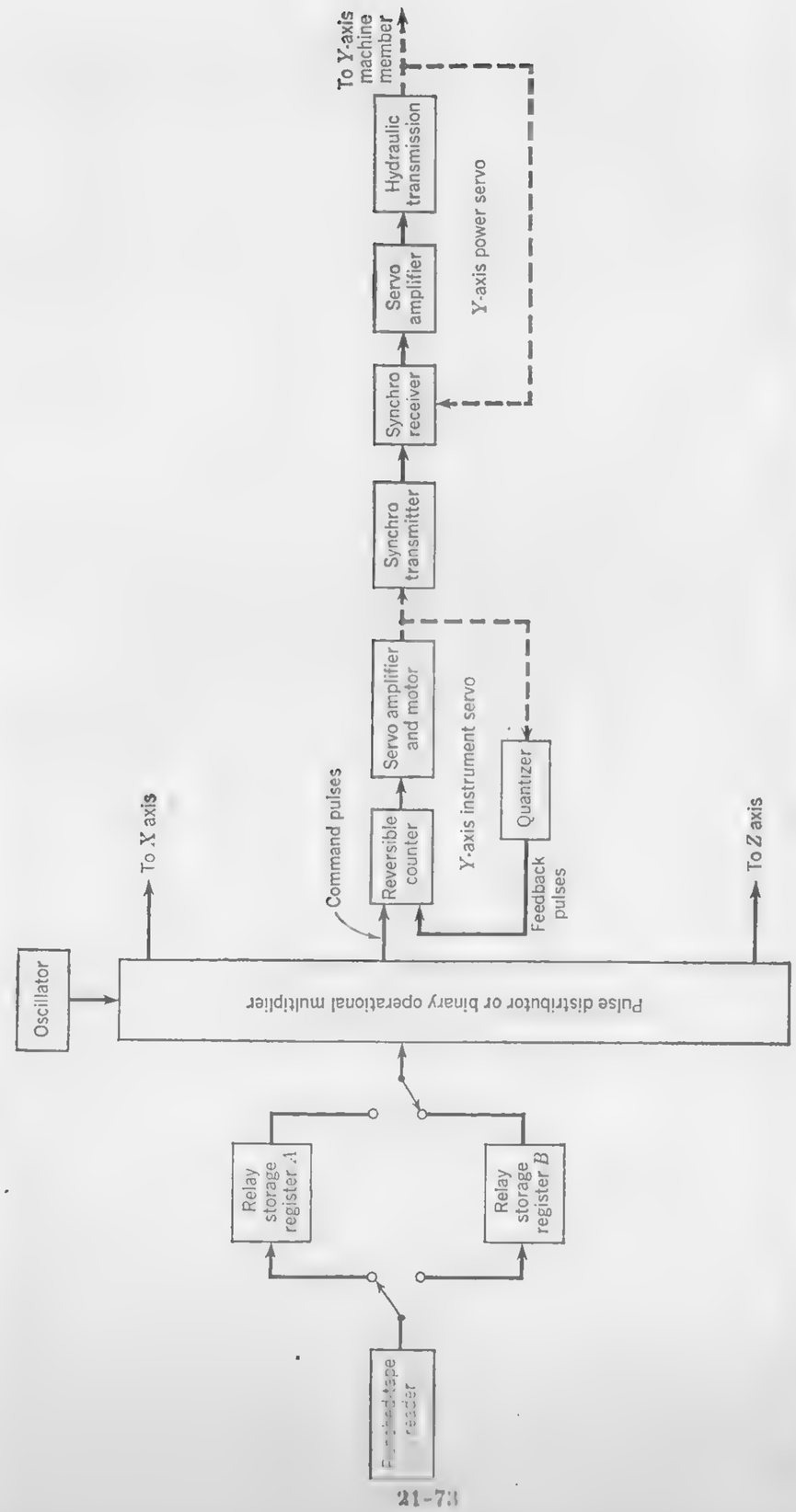


FIG. 21.37. Schematic diagram of the numerically controlled milling machine built by MIT. The system uses digital operational techniques to produce the effect of linear interpolation between specified points.

k for which the coefficient of 2^{10-k} in the binary expression for Δx is 1. Thus the number Δx in parallel binary form may be used to control gates directly to select exactly Δx pulses during the interval. The fact that no two noncarry pulses occur at the same time allows a simple mixing of the pulses from each stage, without possibility of overlap.

Two additional sets of gates, Δy and Δz, operating on the same noncarry lines, permit the generation of three separate pulse trains for controlling the three axes of the machine. The action of the circuitry is such that, if the input pulse rate is uniform, each of the three resulting pulse rates will be nearly uniform over the entire 2-sec interval. Even if the input rate is not uniform, the output pulse trains will

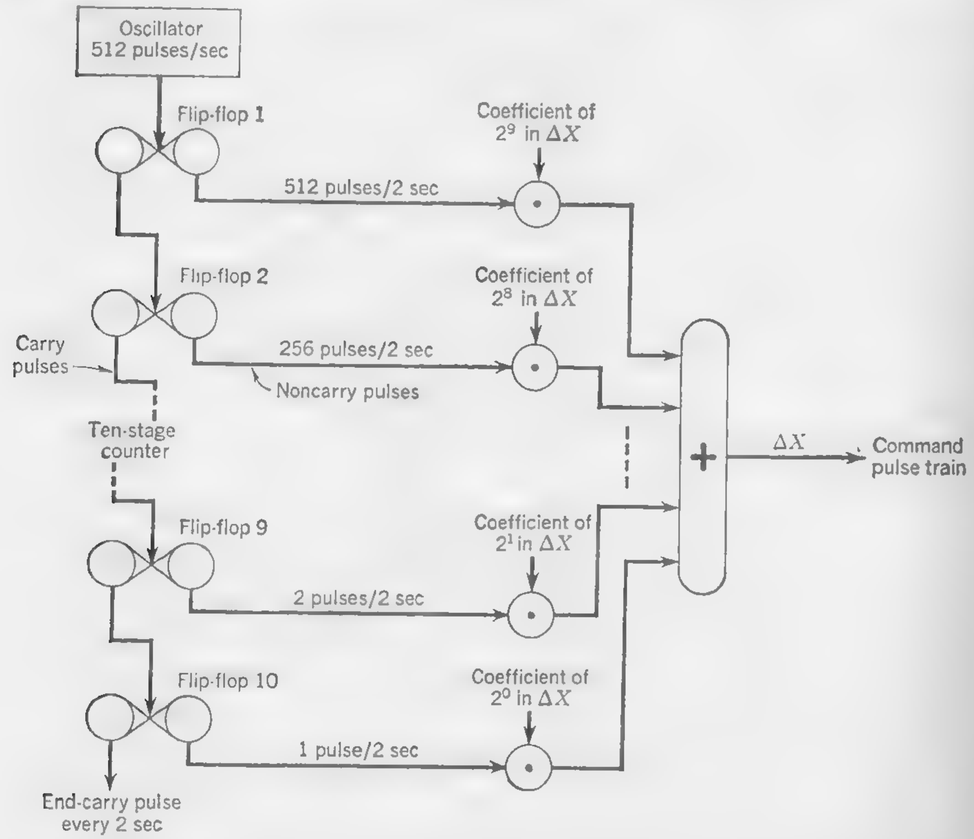


FIG. 21.38. Pulse distributor, or binary operational multiplier, as used in the MIT numerically controlled machine.

always bear approximately the correct ratios to each other. The result, then, is a set of command-pulse trains for the three machine motions describing essentially a straight line in space. (For further discussion of the accuracy of such techniques, see Art. 21.5.3 on binary operational multipliers.)

The quantities Δx, Δy, and Δz referred to above represent the number of 0.0005-in. units of motion required along their respective machine axes in one time interval. With an input pulse rate of 512 pulses per second and a 10-stage counter as described above, the time interval is 2 sec. In the MIT machine, different time intervals are provided for by the inclusion of seven additional counter stages, any number of which may be interposed between the pulse source and the first stage shown in Fig. 21.38. With one additional stage, for example, the time interval is extended to 4 sec; with all seven additional stages, to 256 sec. Also, as more stages are added, the command numbers Δx, Δy, and Δz may contain more digits. At the maximum distributor

length of 17 stages, a stroke of $2^{17} - 1 = 131{,}071$ units may be called for, corresponding to a distance of 65.5355 in.

The particular time interval desired is specified by the punched control tape. Any required feed rate may be approximated closely with the discrete set of time intervals available by subdividing the straight-line section into a number of shorter sections. Suppose, for example, a distance of 12 in. is to be traversed at a feed rate of 6 in. per min. The total time required is 2 min, or 120 sec. This period may be achieved by using one standard interval of 64 sec, one of 32 sec, one of 16 sec, and one of 8 sec (the total being 120 sec). The total distance is divided proportionally: 6.4, 3.2, 1.6, and 0.8 in. for the 64-, 32-, 16-, and 8-sec intervals, respectively. Feed rates may be varied also by changing the pulse rate into the distributor. Provision is made in the MIT machine for continuous manual adjustment over the range from 60 to 120 per cent of the nominal 512 pulses per second.

Control information, including Δx, Δy, and Δz, is read one block at a time into relay registers in the machine-control system. Two complete registers are provided to make possible continuous machine operation from one interval to the next without pause at the boundary. While one register is actively controlling the machine, information for the next interval is being read and assimilated in the second register. When the boundary between intervals is reached, as signified by an end-carry pulse from the distributor, control is shifted to the newly filled register.

The control tape contains seven hole positions across its width. The first row in each instruction block specifies the length of the pulse distributor, and consequently the time interval. The next row gives the direction of motion along the x, y, and z axes. The remaining rows in the block carry Δx, Δy, and Δz. These quantities are arranged longitudinally on the tape, each in a separate channel, least significant digit first. The number of rows required varies from 10 to 17, depending upon the distributor length. Blocks are separated by a special end-of-block row.

The numerical information occupies only three of the seven channels across the tape. The remaining four channels are coded to permit redundancy checking of the tape-punching and -reading operations. In each row of hole positions, a single error can be detected and corrected, and a double error can be detected. Additional checking circuits are included to monitor the operation of most other major portions of the system.

Essentially the same interpolation technique has been applied to the control of a three-dimensional cam-milling machine (Fig. 21.39).[71] The machine is of lathelike configuration, in that the workpiece rotates about an axis as the cutter moves radially and axially with respect to it. The axial motion of the cutter is geared to the work spindle, tending to produce a helical path over the surface of the work as in cutting a screw thread. The cross-slide motion is servoed to vary the radius of the helix continuously in accordance with information supplied by a punched tape. Although the resulting contour is three-dimensional, by virtue of the fixed axial feed only two motions need to be independently controlled: the angle along the helix and the radius of the helix.

The machine utilizes a binary operational multiplier similar in logical function to that of the MIT machine. In the cam machine, however, input pulses for the multiplier are obtained from a quantizer coupled to the work spindle, rather than from an oscillator. Output pulses act as command pulses for the servo varying the radial position of the tool. In effect, then, the radius of the part is controlled as a function of the angle of rotation. In the same terms, the MIT machine may be considered to control three linear motions as a function of time, since the multiplier input is a pulse source having time as its base.

Analogously, operation of the cam machine is divided into *angular* intervals instead of *time* intervals. The spindle quantizer is geared to generate 256 pulses for each 5 deg of cam rotation. An 8-stage multiplier, therefore, produces an end-carry pulse every 5 deg; a k-stage multiplier, every $5(2)^{k-8}$ deg. The length of the multiplier, and consequently the size of the angular interval, is controlled by the input tape. In the original machine, intervals of 5, 10, 20, and 80 deg are available.

Multiplier gates are set in accordance with the number of command pulses to be

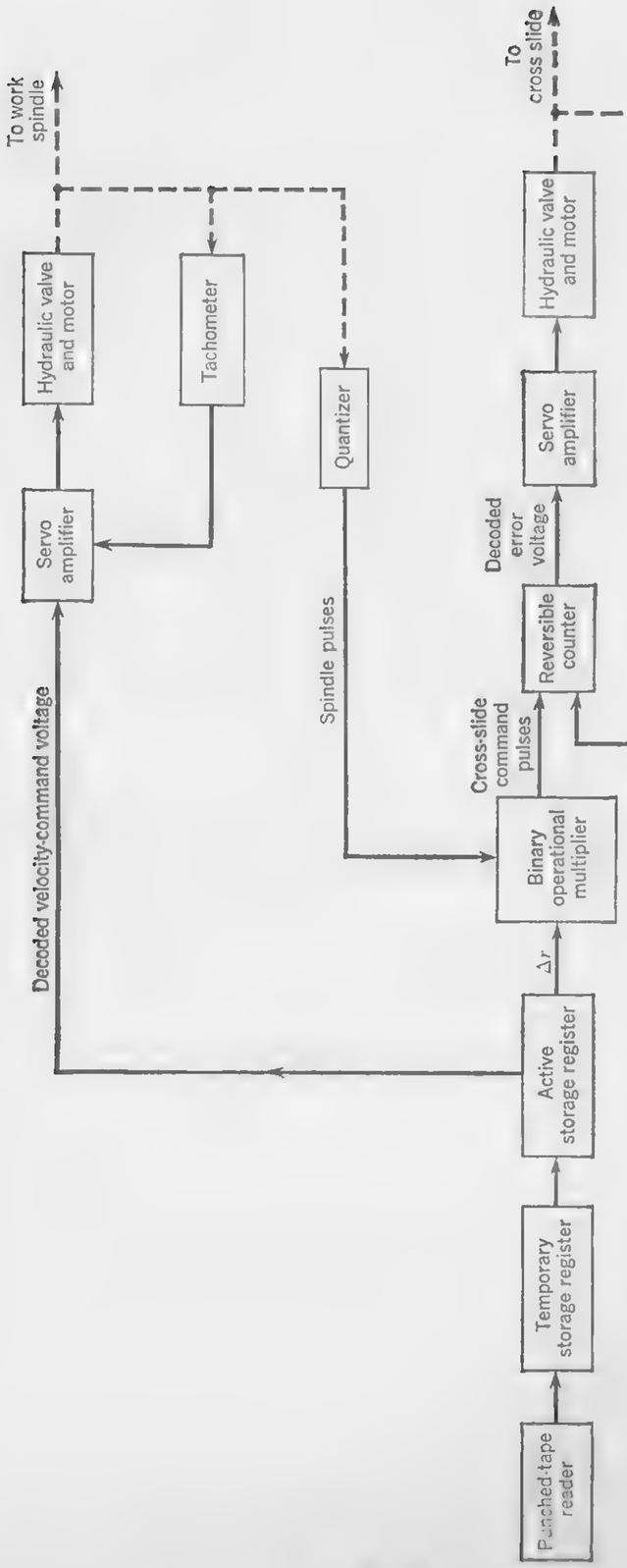


Fig. 21.39. Schematic diagram of a numerically controlled cam-milling machine for cutting three-dimensional cams.

produced in an angular interval. Each such pulse calls for a radial tool motion of 0.0002 in. (The servo which provides this tool motion, using a quantizer for feedback and a reversible counter for error sensing, is described more fully on page 21–66.) If, for example, a change in radius of 0.03 in. is required in 5 deg, 150 pulses must be selected from the 256 pulses produced by the spindle quantizer as the cam rotates through the 5 deg. This selection results if the multiplier gates are opened for those stages whose corresponding digits in the binary representation of 150 (10010110) are 1. Distribution of the 150 pulses is approximately uniform over the 5-deg interval. Consequently, the over-all effect is a linear interpolation of radius as a function of angle. As a result of the cylindrical coordinate system, the actual curve produced is an arc of an Archimedean spiral.

As in the MIT machine, control information is read from punched tape into storage registers arranged to permit continuous operation of the machine. A block of information enters by way of a temporary-storage register and is transferred to an active-storage register when the angular interval to which it applies is reached. Both registers are composed of flip-flops. As compared with the scheme of alternating registers, this system appears to simplify the required switching network, although it perhaps makes greater demands on switching speeds.

In at least one other digital machine-control system, magnetic-core registers perform the buffer-storage function. In this particular system, a magnetic-tape link is interposed between the machine proper and the control equipment which provides the interpolation and digital-to-analog conversion.[91]

Data Storage. Analog Recording on Magnetic Tape. An illustration of the storage of information for machine programming purposes is provided by the so-called "record-playback" process. In this system, the motions of a machine are recorded on magnetic tape as an operator guides the machine, either manually or from templates, through the complete sequence of operations required to produce a part. Then on subsequent parts of the same type, the machine is controlled by the tape record.

Magnetic-tape recording is also used to permit isolating the machine-control equipment, in both time and space, from more complex control functions such as interpolation and digital-to-analog conversion. For example, the final digital-to-analog conversion in a numerically controlled machine system can take place at a low-power instrument servo level.[91] This servo, equipped with synchros or other analog instruments, provides signals to be recorded on magnetic tape which are suitable for controlling the high-power feed servos. This application involves the same basic elements, in so far as data recording and regeneration are concerned, as does the record-playback process. The techniques discussed below are applicable to either problem.

The compactness and relative cheapness of magnetic tape make it attractive in many applications as a data-storage medium. A handicap when recording information in analog form, however, is the nonlinearity of the recording and reproducing process. One of the systems developed to minimize this difficulty involves the recording of phase-modulated synchro data.[114,77] Figure 21.40 shows a simplified version for control of a single motion.

The motion to be recorded, either linear or rotary, is coupled to a synchro control transformer, whose stator is supplied from a three-phase reference source. The voltage induced in the rotor winding is of constant magnitude, but the phase of the voltage with respect to any of the references varies linearly with the angular position of the rotor. One full revolution corresponds to a phase change of 360 deg. The rotor voltage and one of the reference voltages are recorded on magnetic tape as continuous signals. Since the desired information is conveyed solely by the relative phase of these two signals, amplitude distortion of the waveforms is not serious as long as the original phase relationship can be reconstructed.

The recorded motion is reproduced by a servo provided with a similar control transformer as a feedback device. Three-phase excitation for the stator winding is derived from the recorded reference by means of a three-phase amplifier. The phase of the resulting rotor voltage is compared in a phase discriminator with the phase of the recorded rotor signal. Any difference in phase, representing misalignment of synchro rotors, produces an error signal which is amplified and applied to the feed-drive motor.

Additional motions can be handled in the same way. An advantage of this sytem is that only one additional information channel is needed for each motion, the reference being common to all motions. A separate tape track may be used for each information channel, or several signals may be superimposed on the same track by modulating the signals onto suitable carriers. In record-playback applications, the same synchro and tape-transport equipment can, with appropriate switching, serve in both the recording and the reproducing operations. When the tape link is used for time and space isolation, however, separate equipment must be provided.

Static accuracy of the data-transmission system depends upon tape speed, data-carrier frequency, synchro accuracy, and effective dimensional resolution on the tape. By the latter is meant uncertainty in the relative position on the tape of signals in

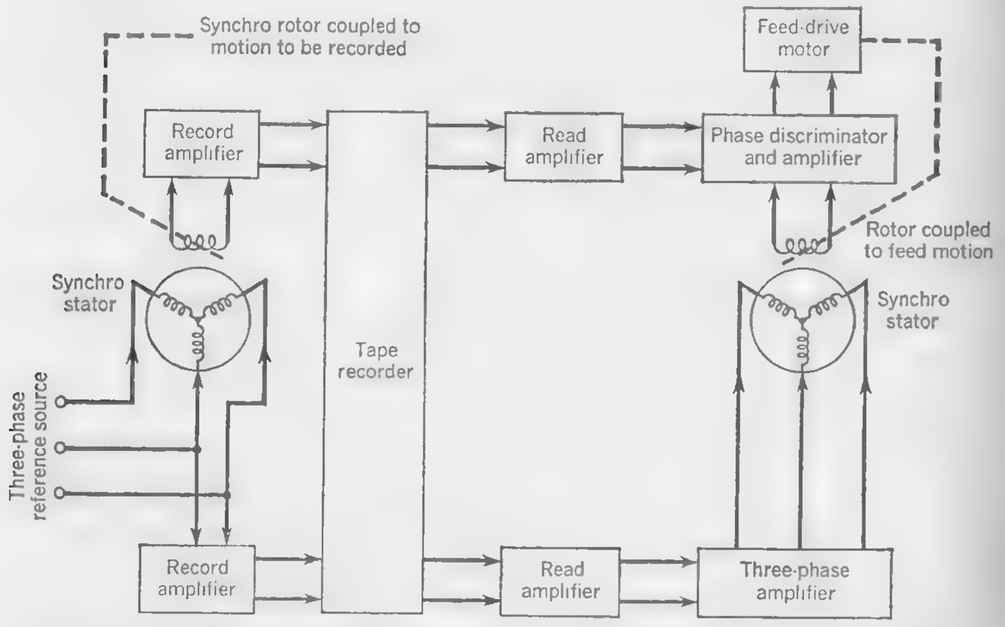


FIG. 21.40. A phase-modulation system for the magnetic recording of synchro data.

different channels due to misalignment of reading and recording heads, tape skew and jitter, and tape granularity and compliance. The effect of this uncertainty is similar to the effect of synchro inaccuracy. The distance one cycle of the data carrier occupies along the tape is v/f, where v is tape speed and f is carrier frequency. If r is the minimum resolvable distance on the tape, rf/v is the uncertainty in synchro alignment as a fraction of a revolution. For example, if $f = 60$ cps, $v = 3\frac{3}{4}$ in. per sec, and $r = 0.0005$ in., the possible synchro error is 0.008 revolution, or about 3 deg. In general, r can be smaller—and therefore this effect less significant—in record-playback operation, where the same tape-transport mechanism and magnetic heads are used for both recording and reproducing. However, in many applications, the data-carrier frequency must be well above 60 cps to achieve the dynamic performance required, further magnifying the effect.

Another recording system which has been applied to machine tools uses a comblike device to lay down a series of parallel magnetized lines along the length of the tape.[64,66] The comb is attached to the motion to be recorded, and as it moves across the tape it causes a lateral shifting or skewing of the lines (see Fig. 21.41). This shift is sensed in the reproducing operation by a rotating scanning head. The position-measuring equipment for the motion being reproduced is a similar arrangement of a precision magnetically ruled tape and scanning head. Signals produced by the two scanning heads are compared in phase, and any difference is amplified to actuate the feed-drive motor. Over-all accuracy depends directly on the lateral tape resolution,

but the system allows control to be maintained during slowdown and stopping of the tape.

Contouring from a Model. Another class of computing functions is found in contouring systems which have been developed to guide the operation of a machine from

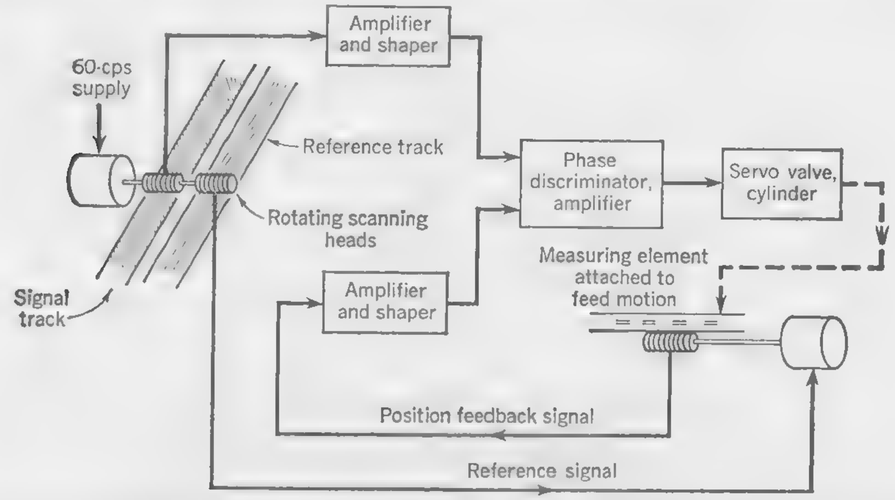


FIG. 21.41. An alternate system for recording analog position information on magnetic tape.

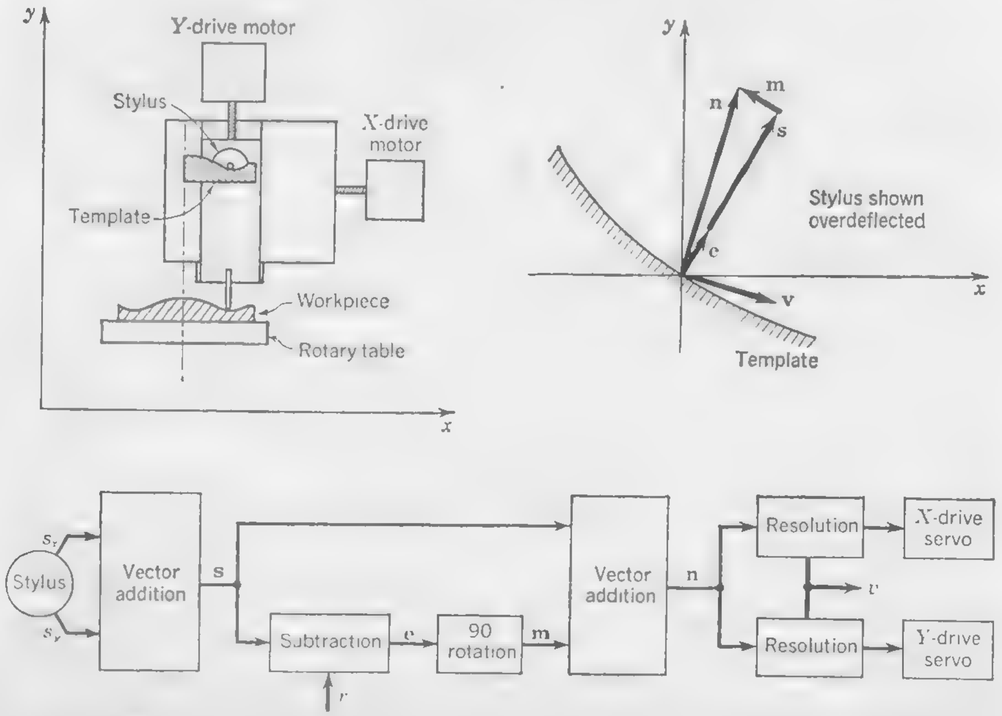


FIG. 21.42. Basic functions involved in one type of two-dimensional 360-deg profiling system. A typical application is the vertical-lathe control illustrated.

a model or a template. Those systems which make use of signals produced by an electrical stylus frequently involve a series of vector manipulations. Typical of the "360-deg profilers" is the system illustrated in Fig. 21.42. In this vertical-lathe application, a slide carrying the stylus is servoed in two directions so that the stylus

tip follows the edge of a template. Since the tool also is mounted rigidly to the slide, it duplicates the motions of the stylus, reproducing the template contour in the workpiece. The nature of the control system is such as to steer the stylus completely around the template, maintaining constant tangential velocity. The same basic system may be applied to any type of machine which requires contouring control of two orthogonal motions.

The stylus employed contains two transducers mounted at right angles to each other. One transducer is aligned with the x axis and produces a signal s_x proportional to the component of stylus deflection in the x direction. Correspondingly, the other transducer responds to deflection in the y direction and produces a signal s_y. These two components of stylus deflection are combined by vector addition. The resultant vector **s** represents both the magnitude and the direction of stylus deflection. If friction and inertia forces acting on the tip are low, the stylus deflection will be nearly normal to the template at the point of contact. The direction of tangency therefore can be obtained by rotating **s** 90 deg.

Systems of this type must operate about a nominal stylus deflection of constant magnitude in order to sense departure from the template, as well as to give continuous indication of the direction of tangency. Ideally, in the steady-state tracing of a straight section of the template, the stylus would be deflected exactly the nominal amount, and the x and y servos would be provided with velocity commands proportional to the y and x components of deflection, respectively. When the steady-state situation is disturbed, as at a corner or over curved sections of the template, deviation of stylus deflection from nominal causes the servo commands to be modified so as to tend to restore the condition of tangency. If the stylus is deflected too much, it should be steered away from the template; if too little, toward the template.

This action is obtained by comparing the magnitude of the stylus deflection **s** with the nominal deflection r (see Fig. 21.42). Subtraction of r from s results in the error vector **e**, representing the over- or underdeflection. For clockwise tracing, this vector is rotated counterclockwise 90 deg and added to **s**, producing **n**. The direction of tracing then is taken normal to **n**, with an independently specified velocity v. Thus the velocity commands applied to the x and y feed drives are $(v/n)\ n_y$ and $-(v/n)\ n_x$, respectively.

The vector diagram in Fig. 21.42 illustrates a case in which the stylus deflection is too great, indicating that the stylus body is too close to the template. The vector **e** is therefore in the same direction as **s**. Rotation by 90 deg and addition to **s** produces **n**, which leans back from **s**, away from the direction of tracing. The velocity command vector **v**, normal to **n**, is therefore away from the template to correct for the overdeflection.

In one embodiment of this system, the vectors are represented by 60-cps voltages.[24] The magnitude of the voltage corresponds to the magnitude of the vector, and the phase angle with respect to a reference voltage corresponds to the direction of the vector. The stylus employs a movable-core transformer to produce the two voltages s_x and s_y. The phases of these voltages, initially the same as the phase of the transformer primary excitation, are shifted 45 deg in opposite directions. Being 90 deg apart, then, the voltages may be added directly to obtain the vector **s**. The error vector **e** is produced by a modulator whose carrier excitation is **s** and whose unmodulated input is the difference between a rectified **s** and a d-c signal r representing nominal stylus deflection. The indicated 90-deg rotation is performed by a Miller phase shifter, and the vector addition is again a direct combination of the voltages, since **s** and **m** are always 90 deg out of phase. Resolution into v_x and v_y is performed by phase-sensitive demodulators. The voltage representing **n** serves as reference excitation after being shifted +45 deg for one demodulator and −45 deg for the other. Feed velocity v, represented by a voltage in phase with the stylus excitation, acts as the input signal to be demodulated in each case.

In a variation of the basic system illustrated in Fig. 21.42, the x and y stylus transducers are excited by reference sources 90 deg out of phase.[26] Since the resulting signals s_x and s_y are then in phase quadrature, they may be added without further phase shifting to produce the deflection vector **s**. Also, in this system, the equivalent of the

vector **n** is produced by shifting the phase of **s** an amount proportional to the magnitude of **e**. The shifting is accomplished in a bridge circuit having a saturable reactor in one arm. The reactor is biased by a d-c signal obtained by subtracting r from a rectified and filtered **s**.

21.5.2. Equation-solving Controls. *Types and Applications of Equation-solving Controls.* In a wide variety of machine and process-control equipment, signals which actually operate the valves, motors, and other control devices are derived from a group of input signals by continuous solution of mathematical equations defining the desired behavior. Systems of this nature are extensively used for aircraft flight and navigation control, gunfire control, and bombing control, as well as for control of chemical, petroleum, and other processes.

From the standpoint of computing functions, some of the most complex of these controls are found in military guidance and weapons-control systems. The mathematical problems involved are principally those of solving the equations of motion of one or more bodies, with associated coordinate transformations. Statistical prediction and smoothing and the solution of algebraic equations may also be required. Additions, multiplications, integrations, and trigonometric resolutions are therefore common. The necessity for handling these problems in real time is often a further complicating factor.

The computing equipment used in these controls has been predominantly of the analog variety. It may be further classified as (1) mechanical, (2) electromechanical, and (3) electronic.[54] In the mechanical type, the analog quantity is usually shaft rotation or a mechanical displacement; and the computing is done typically by differential gears, ball-and-disk integrators, and linkages. The electromechanical type makes extensive use of instrument servos to drive potentiometers, resolvers, and tachometers. The relative ease with which multiplications, trigonometric resolutions, and other nonlinear operations can be handled by this type of equipment has led to widespread usage. It is usually faster than the mechanical type, although perhaps not so accurate. Electronic computing equipment is well adapted to problems which are essentially linear and which require integration with respect to time only. Its potentially high operating speed in these applications is attractive. However, all-electronic equipment for performing most nonlinear operations is too complex to permit its use except where a high premium for speed can be justified.

The technique adopted in any particular case depends upon the accuracy and computational operations to be performed and upon the nature of the input and output signals. Under certain conditions, the use of a digital computer becomes an obvious possibility—for example, if extreme accuracy of computation is required, or if the input information is available naturally in digital form. A digital computer may also be attractive if a number of logical decisions must be made, with associated changes in mode of operation.

The next paragraphs describe a digital aircraft-navigation system to illustrate the application of digital techniques to equation-solving controls, and also to illustrate the characteristics of guidance and weapons-control systems in general. Succeeding sections discuss the computing functions involved in controls for the continuous-process industries, and the application of analog equipment to control the generation and distribution of electric power.

A Digital Aircraft-navigation System. Digitac (digital tactical automatic control) is the name given to a digital computer and associated input-output equipment developed for automatic aircraft-navigation and weapons control.[19,12] The principal function of the computer is to calculate periodically the position of an aircraft, and from this to generate the steering signal necessary to guide the aircraft over a prescribed course. Other computer functions, having to do mainly with weapons-control aspects of the system, have not been revealed because of military security restrictions. Figure 21.43 is a block diagram of the complete system showing how a computed steering signal is applied to the autopilot which controls the aircraft directly.

Position of the aircraft is determined by reference to three ground transmitters, using a hyperbolic method similar to Loran. The difference in propagation times of pulses simultaneously emitted by two stations establishes a hyperbola on which the

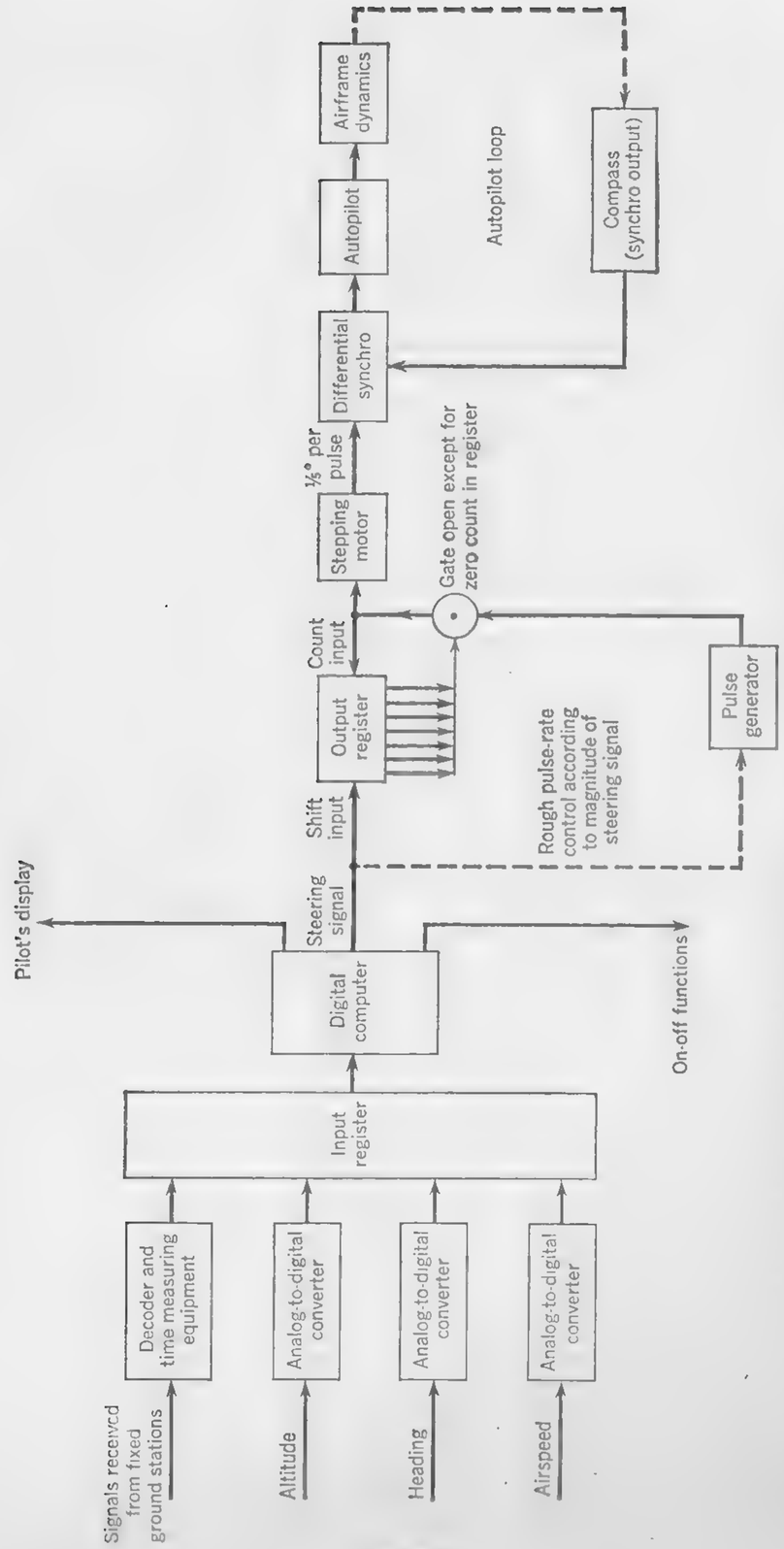


FIG. 21.43. Application of a digital computer to aircraft-navigation control. (Digitac system.)

aircraft is known to lie. A second hyperbola is established by the time difference between pulses from another pair of stations. The intersection of the two hyperbolas (actually hyperboloids of revolution), together with a measured altitude, gives the location of the aircraft. The Digitac equipment is reported to be capable of measuring the required time differences to within ± 0.02 μsec. Altitude and other input information, consisting of heading and air speed, are obtained from shaft positions by binary-coded digitizers read on demand by the computer.[127]

The computer itself is of the general-purpose type utilizing a magnetic drum for nonvolatile storage. Space is provided for 768 instructions, 192 constants, and 96 words of working storage. A 6-word fast-access circulating register is also included. Computation is carried out serially at a basic clock rate of 100 kc. The word length of 17 digits provides a precision of 16 bits plus sign in the case of numbers, or the equivalent of one two-address order in the case of commands. A floating-reference

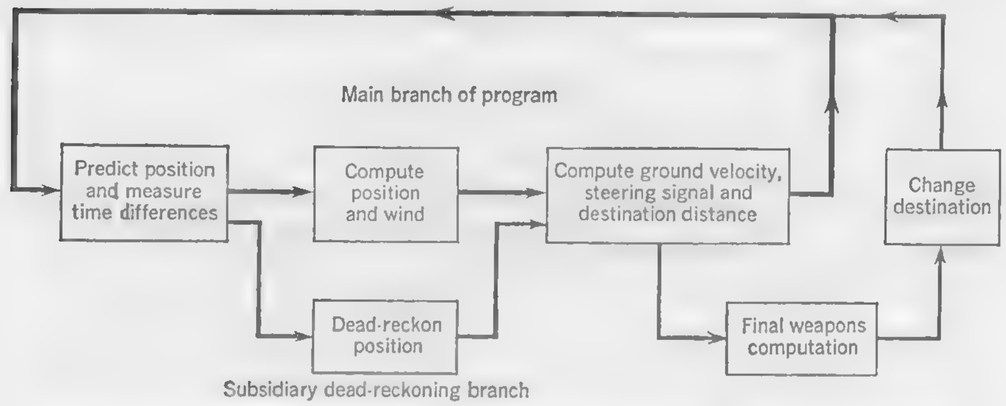

Main branch of program
Predict position and measure time differences
Compute position and wind
Compute ground velocity, steering signal and destination distance
Change destination
Dead-reckon position
Final weapons computation
Subsidiary dead-reckoning branch

FIG. 21.44. Flow chart of navigational problem as handled in the Digitac system.

addressing scheme leads to a very compact command structure, simplifies the equipment, and facilitates optimum-access coding.

A typical computational program as employed in flight tests of this equipment is indicated by the flow chart of Fig. 21.44. The cycle begins with a computation of predicted present position on the basis of (1) the previous computed position, (2) previous computed wind, (3) current measured air speed, and (4) the time elapsed since the previous position computation. Then a measurement of the two time differences is made. If these differences are within bounds and are reasonable with respect to the previously measured differences, the main branch of the program is followed. In this branch, present position is computed by means of an iterative routine from the intersection of the two hyperbolas defined by the time differences. This computed position is combined with the predicted position, giving equal weight to both, to arrive at a smoothed present position.

A rough value for wind is computed on the basis of this smoothed position and combined with the previous wind value to give a smoothed wind. In the wind smoothing, the old value is weighted 31 times as heavily as the new rough value. In the next step of the program, ground velocity is computed. From this and the desired course, a steering signal is developed. Also, the remaining distance to the destination is checked. If the destination has not been reached, the main cycle starts again; if the destination has been reached, either a change in destination is made or the final weapons computation begins.

If the time differences as measured at the beginning of a cycle are found not to be usable, the dead-reckoning mode is entered, bypassing the hyperbolic-position computation and the wind updating. The predicted position alone is used in the subsequent step to compute ground velocity and steering signal. The normal full cycle requires about 0.5 sec to execute and involves about 360 operations; the dead-reckoning mode requires only about 0.3 sec. In order to perform routine weapons calcula-

tions (which are not indicated in Fig. 21.44) without consuming additional time, the computer follows the dead-reckoning mode every 32 cycles. The 0.2 sec thus gained is sufficient time for the computations required.

The steering signal generated every 0.5 sec represents the change in heading required to reach a specified objective. As indicated in Fig. 21.43, this signal is shifted into an output register containing 7 digits plus sign.[36] This register controls the flow of pulses from a pulse generator to a bidirectional stepping motor. The same pulse train also causes the output register to count toward zero. When zero is reached, indicating that the stepping motor has received the required number of pulses, the register shuts off the flow of pulses. If zero count is not reached within the 0.5-sec interval, a new steering signal is shifted in anyway, without regard for the remainder of the old signal.

The stepping motor drives a synchro differential which is inserted in the normal synchro link between compass and autopilot. Its effect is to add to or subtract from the compass signal and thereby modify the effective heading command. The stepping motor itself rotates $\frac{1}{50}$ revolution for each pulse received, rotation being in the direction of the required change as indicated by the sign of the output register. A gear ratio of 36:1 between motor and synchro results in a heading change of $\frac{1}{5}$ deg for each pulse. In order to increase stability of the navigation loop and minimize overshoot, the pulse rate is made roughly proportional to the magnitude of the steering signal, ranging approximately from 4 to 15 cps.

Coupling the computer to the autopilot loop in this manner makes less stringent demands on speed of computation than if the computer were placed directly in the autopilot control loop. In the arrangement of Fig. 21.43, the effective computer bandwidth need be sufficient to handle only the relatively slowly changing navigation signals. Aircraft stability is not seriously disturbed since the transmission of analog information around the autopilot loop is not affected. In flight tests of the system, the cycle time of 0.5 sec required for the navigation computation appeared to cause no difficulty in obtaining adequate stability of the guidance loop.[51] In general, however, when digital computers are applied to real-time control problems involving feedback loops, the effect on stability of discrete data sampling should be considered.

Continuous-process Controls. The control equipment for a chemical or petroleum process consists typically of a large number of more-or-less independent controllers, each of which performs some relatively simple function such as the regulation of a temperature, pressure, or flow. In some cases, the resulting variables are passed along to serve as reference levels for subsequent controllers. In other cases, the outputs of several controllers and measuring elements may be combined according to some mathematical relationship for further control or for indication and metering purposes. Coordination of the many separate control loops is provided principally by the human operators who continuously monitor the important process variables and adjust the regulating levels of the individual controllers in order to obtain the desired over-all results.

The bulk of controllers in use today are pneumatically operated; however, there is a growing trend toward electrical controls in new plant construction, particularly in critical applications. Pneumatic controllers are relatively simple and have given excellent service for many years. But as processes become more complex and the plant layouts larger physically, the disadvantages of pneumatic controls become more serious. A satisfactory air supply and distribution system for an extensive pneumatic control system is costly, in both initial installation and subsequent maintenance. Time lags associated with the transmission of information through long air lines definitely limit the dynamic performance that can be achieved in a feedback-control system. Because of the absence of these lags in electrical systems, greater control sensitivities are possible and more effective use can be made of derivative or anticipatory control. In general, electrical systems are considered to be more flexible and better adapted to the more complex control functions which will be found desirable as the dynamics of processes become better understood.

The elements of a simple electrical controller are shown in Fig. 21.45. Its principal functions are to:

1. Receive signals from a measuring device
2. Indicate or record the variable of interest
3. Provide signals at a higher energy level for control purposes

The electrical signal from the measuring element is first converted to a shaft rotation in a null-seeking system which balances the incoming signal against that developed by an instrument coupled to a motor shaft. The motor drives a recording pen or other indicator, and it may also drive a second instrument or retransmitter. The latter provides signals for actuating a valve or other device by means of which the control loop is closed.

Considerable flexibility is provided by this arrangement in that either the feedback instrument or the retransmitter or both can generate nonlinear functions of the motor position. The instruments themselves may be nonlinear—resolvers or nonlinear potentiometers—or they may be linear devices coupled to the motor by means

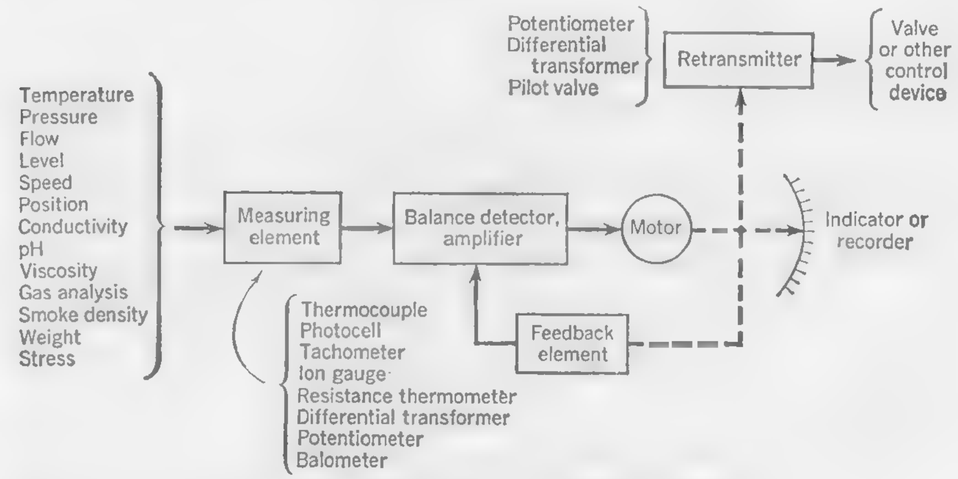


FIG. 21.45. Elements of an electrical controller.

of specially shaped cams. By virtue of these possibilities, simple computing functions can be performed by the controller.

Suppose, for example, the flow of some process variable must be recorded and a proportional signal developed for control purposes. A common method for metering flow makes use of the principle that the pressure drop across an orifice is proportional to the square of the flow through it. Pressure drop is measured and the square root then taken as an indication of flow. The square-root operation required in this scheme can be performed by the controller if the feedback instrument develops a signal proportional to the square of the motor-shaft rotation. The motor shaft, thus representing the square root of the measured quantity, drives the recorder pen arm directly to record flow. The desired control signal is obtained from a linear transducer coupled to the motor.

In a system of several controllers, addition and subtraction can be performed by the balance detector and amplifier (Fig. 21.45) or by series connection of the voltage sources to be combined. Multiplication can be performed if the instrument (such as a potentiometer) connected to a shaft representing one factor is supplied by a signal representing the other factor, instead of by a reference signal. Figure 21.46 illustrates the use of three such controllers to compute and record the heat absorbed by a heat exchanger.[65]

Completely automatic control of an entire process would require not only the regulation of environmental variables such as temperatures, pressures, and flows, but also some means for determining at what level they should be regulated. Presumably, the composition of the raw materials and the composition of the products at strategic points in the process should be taken into account, along with the many other factors

which affect the end result. The computational functions required in such a comprehensive control system would be considerably more complex than those found in the process controllers of today, but possibly well within the present state of the computer art. Instruments are now becoming available which can analyze the composition or quality of a material on a continuous basis, removing one of the obstacles to complete control. In most cases, the principal technical problem remaining is the development of mathematical models which represent the process dynamics with sufficient accuracy to serve as a satisfactory basis for automatic control.[148]

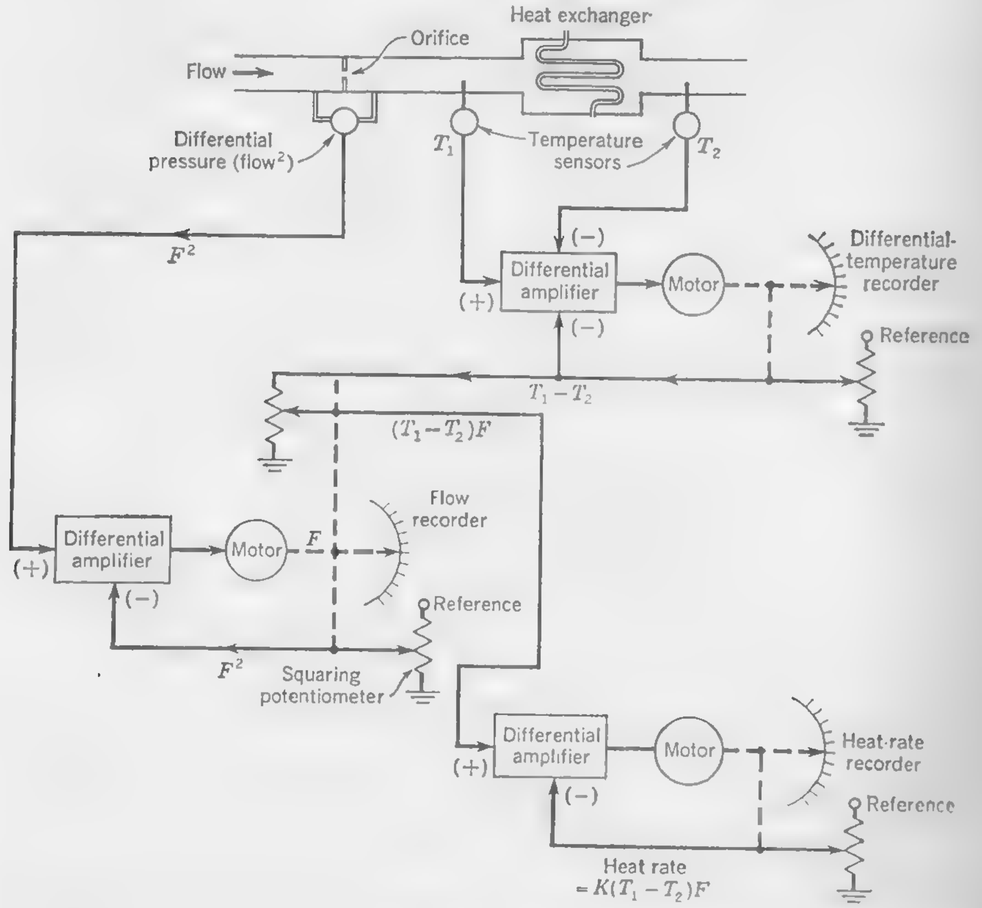


FIG. 21.46. Combination of three controllers to compute heat absorbed by a heat exchanger. (Adapted from Fig. 2, Ref. 65.)

Electric-power Generation. Many possibilities exist as to the exact division of any given load among the several generating units of a large power system. The optimum choice depends mainly on economic considerations—for example, the cost of fuel at the different stations, the efficiency of the individual units within a station, and the transmission losses between the stations and the loads. In some cases, computers are now being applied to monitor continuously the system power requirements and to control the generating units on some simple economic basis.[58,33] One such control system employs more-or-less conventional electronic-analog-computing techniques.[103,35]

In a system consisting of N generating stations, the most economical load division occurs when the equation

$$\frac{dF_n}{dP_n} + \lambda \frac{\partial P_L}{\partial P_n} = \lambda \qquad (21.49)$$

is satisfied for each station n. In this equation, P_n is the power produced by the nth station, F_n is the production cost per unit time at the nth station, and P_L is the total power loss of the transmission system. The quantity λ, which is common to all N equations, may be interpreted as the actual cost per unit of energy delivered by the system.

The total transmission loss is commonly represented by

$$P_L = \sum_{m=1}^{N} \sum_{n=1}^{N} P_m P_n B_{mn} \qquad (21.50)$$

where the constants B_{mn} are characteristics of the transmission system. Consequently,

$$\frac{\partial P_L}{\partial P_n} = 2 \sum_{m=1}^{N} B_{mn} P_m \qquad (21.51)$$

The derivative dF_n/dP_n is taken to be proportional to the incremental heat rate dH_n/dP_n:

$$\frac{dF_n}{dP_n} = K_n \frac{dH_n}{dP_n} \qquad (21.52)$$

where K_n is the cost of fuel per unit of heat energy. Substitution of Eqs. (21.51) and (21.52) in Eq. (21.49) leads to

$$\frac{dH_n}{dP_n} = \frac{\lambda}{K_n}\left(1 - 2 \sum_{m=1}^{N} B_{mn} P_m\right) \qquad (21.53)$$

At the most economical load division, all generating units within a station are operating at the same incremental heat rate, which is equal to that for the station as a whole. That is,

$$\frac{dH_n}{dP_n} = \frac{dH_{n1}}{dP_{n1}} = \frac{dH_{n2}}{dP_{n2}} = \cdots \qquad (21.54)$$

where the second subscripts denote the individual units of a station. The incremental heat rate dH_{nk}/dP_{nk} is generally a known function of generated power P_{nk} for each unit of a station. Also, of course, the total station power is the sum of that produced by all units:

$$P_n = P_{n1} + P_{n2} + \cdots \qquad (21.55)$$

Figure 21.47 shows an arrangement of electronic-analog-computing elements for simulating one station of a power system on the basis of Eqs. (21.53), (21.54), (21.55), and the known relationships $P_{nk} = f\,(dH_{nk}/dP_{nk})$. The indicated summations are performed by resistive networks, and the multiplications by potentiometers and amplifiers. The heat-rate functions are represented by biased-diode networks. The switches associated with the heat-rate functions permit including only those generating units which are available for operation. Lower and upper limits on station production may be set to provide adequate local protection, or for other reasons.

Several station simulators are connected together as in Fig. 21.48 for control of a complete system. Total computed system generation, as obtained by summing the results of all station simulators, is compared continuously with the total power required of the system. Any discrepancy causes a λ servo to increase or decrease, driving the λ potentiometers in the station simulators so as to maintain the proper balance. Control signals for the individual generating units are taken from the outputs of the incremental-heat-rate function generators.

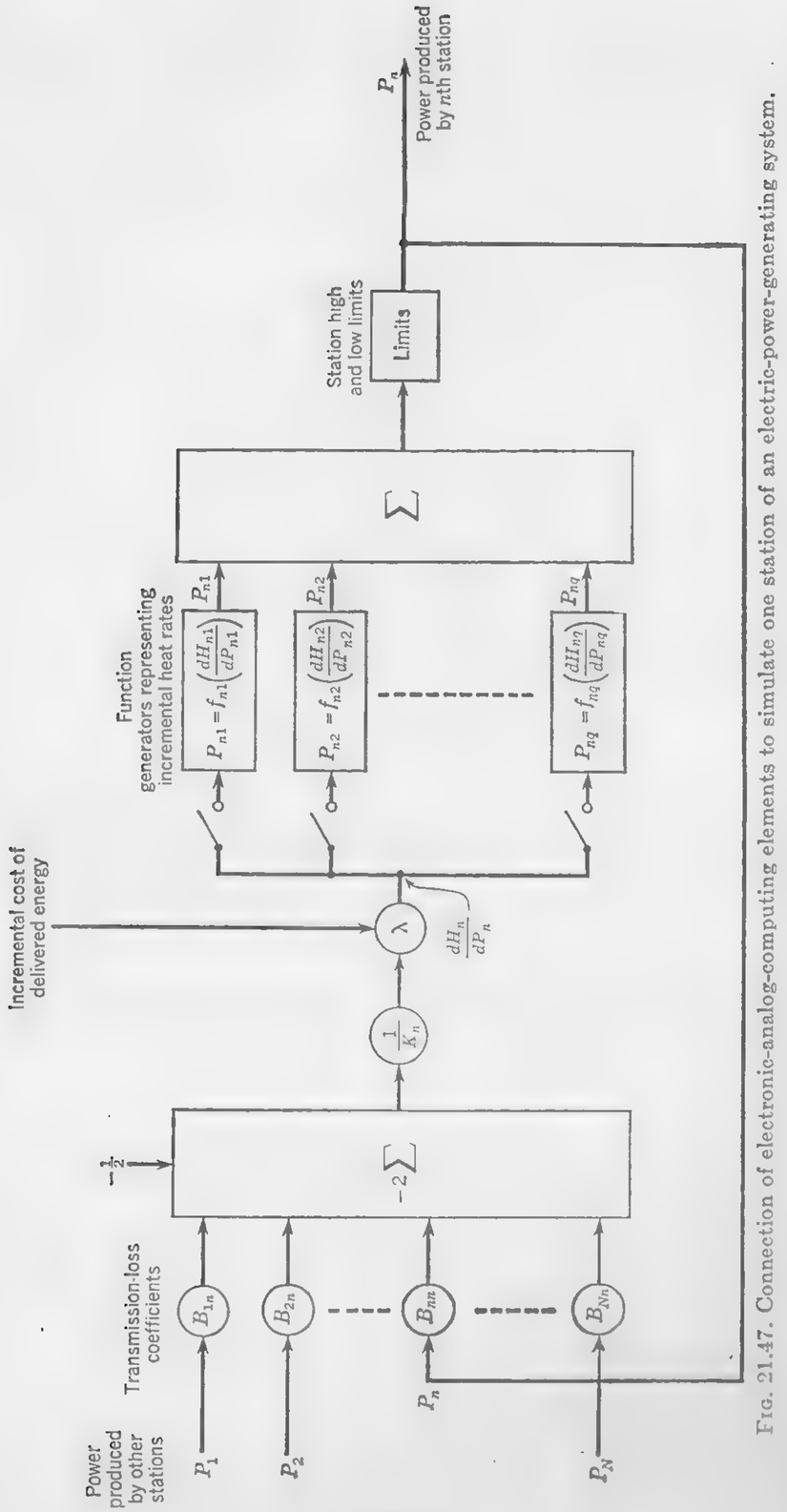


FIG. 21.47. Connection of electronic-analog-computing elements to simulate one station of an electric-power-generating system.

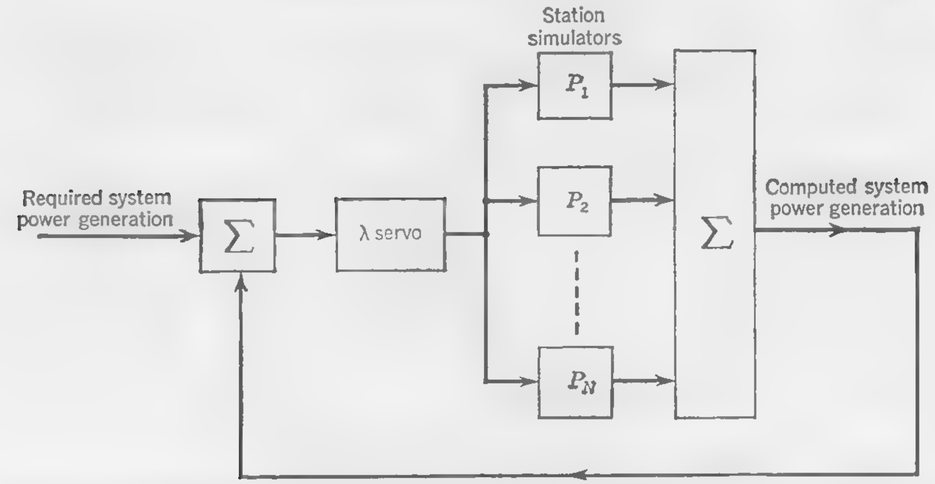


FIG. 21.48. Connection of station simulators for control of system. Power level required of each generator is furnished by station simulators.

21.5.3. Digital-operational Techniques. *General Characteristics of Digital-operational Systems.* The term "digital-operational," or "operational-digital," has been applied to a class of computing techniques which combines some of the characteristics of both digital and analog methods. The over-all computational concepts are those of analog operational systems, in which information flows continuously and on an "as-received" basis from one device or circuit to another. Each such device, which is usually a physically distinct entity, performs some mathematical operation on its input data, such as multiplication, integration, and addition. The digital characteristics of these systems result from the fact that the problem variables are represented in discrete form, as in other types of digital equipment, and the devices which "operate" on them are digital in nature, rather than analog.

Digital-operational techniques potentially allow the simultaneous realization of many of the advantages inherent in both digital and analog approaches. Unlimited computational precision can be obtained, without mechanical or electrical precision, merely by extending the number of digits used to represent the problem variables. At the same time, the functional simplicity of the analog system is retained. The continuous nature of the computation and the ability to operate in real time are of special importance in machine- and process-control applications.[47]

Binary Operational Multipliers and Counters. An important element in digital-operational systems is the binary operational multiplier (Fig. 21.49). This is basically a device for scaling down an input pulse train to produce another pulse train having on the average some specified fraction of the original pulses. As a multiplier, one factor is represented in uncoded form by the number of pulses in the input pulse train, and the other factor by an n-place binary number (less than 1) specifying the fraction of input pulses desired. The resulting pulse train represents the product of the two factors, again in uncoded form.

The operation of the multiplier is described briefly on page 21–72 in connection with

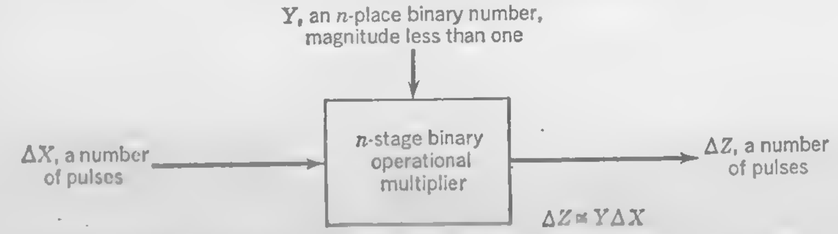

Y, an n-place binary number, magnitude less than one
ΔX, a number of pulses
n-stage binary operational multiplier
ΔZ, a number of pulses
$\Delta Z \cong Y \Delta X$

FIG. 21.49. Relationships involved in a binary operational multiplier.

its application to a digital machine-tool control system. More generally, it may be represented as in Fig. 21.50. In its simplest form, the multiplier consists of an n-stage binary counter and a set of gates connected to the noncarry outputs of the individual stages. The pulse group ΔX is applied to the input of the counter, and the n-place number Y controls the gates. The gate outputs are combined to form the pulse group ΔZ representing the product.

During any period over which ΔX is a multiple of 2^n pulses, the kth stage of the counter produces $\Delta X/2^k$ noncarry pulses. If Y remains constant during this period,

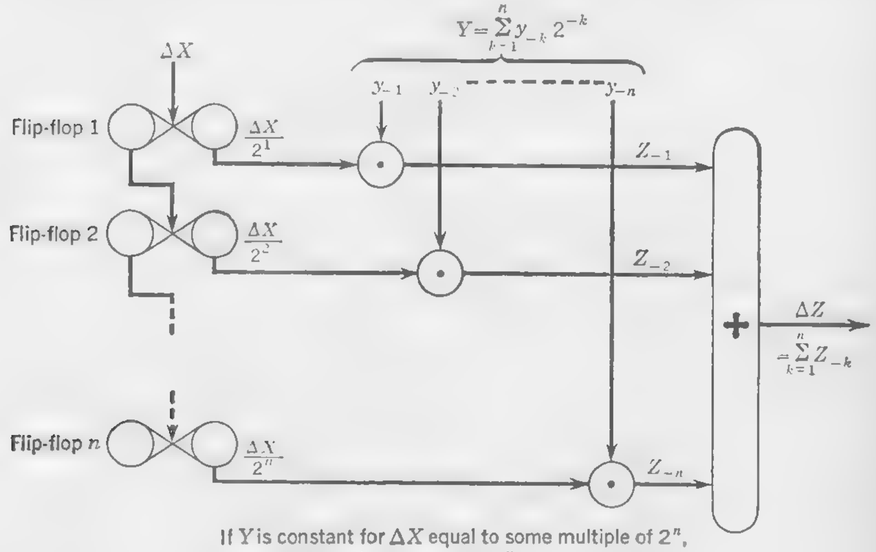


FIG. 21.50. Schematic representation of an n-stage binary operational multiplier.

the kth component of ΔZ is

$$\Delta Z_{-k} = y_{-k}\,\Delta X 2^{-k} \tag{21.56}$$

where y_{-k} is defined by

$$Y = \sum_{k=1}^{n} y_{-k}2^{-k} \tag{21.57}$$

Thus

$$\Delta Z = \sum_{k=1}^{n} \Delta Z_{-k} = \Delta X \sum_{k=1}^{n} y_{-k}2^{-k} = (\Delta X)Y \tag{21.58}$$

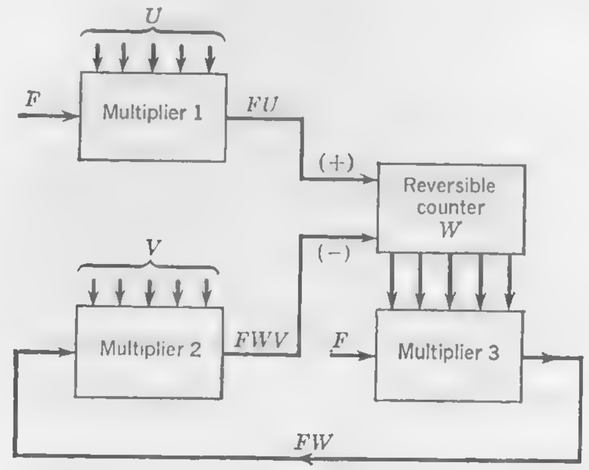
This relationship is exact as indicated when ΔX is a multiple of 2^n and when Y is constant over that interval. In most applications reported to date, these conditions are not always met. However, if Y changes slowly compared with the pulse rate of X, the multiplication is approximately correct. Errors which result under these circumstances are discussed further on page 21–93.

Another major element in digital-operational systems is the counter. In a sense, the counter performs a function opposite to that of the multiplier. By accumulating input pulses, it is capable of producing a coded representation from an uncoded pulse-by-pulse representation. The multiplier, on the other hand, if supplied from a reference pulse source, produces a number of pulses or a pulse rate corresponding to a coded input.

Both devices deal with coded and uncoded pulse groups, which are the two principal

forms of data representation in digital-operational systems. Of the coded class, binary or base-2 codes have been most common, although others are possible. For example, decimal counters could easily be used to accumulate end results if decimal read-out were desirable. Operational multipliers which accept decimal rather than binary numbers are also possible, so that an entire system could be constructed in decimal form.

Illustrative Operational Systems. The combination of three multipliers and a reversible counter to form a divider is illustrated in Fig. 21.51. The dividend U and the divisor V are assumed to be available in coded form as Y inputs to multipliers 1 and 2. A reference pulse rate F is applied to the X inputs of multipliers 1 and 3. As a result, multiplier 1 produces a pulse rate FU, equal to the fraction U of the input rate F. The count W standing in the reversible counter controls multiplier 3 and causes it to produce a rate FW. This in turn is applied to multiplier 2, producing the rate FWV. The counter has two inputs which cause counting in opposite directions.



$$FU = FWV$$
$$W = \frac{U}{V}$$

FIG. 21.51. Connection of binary operational multipliers and a reversible counter for division.

Steady-state equilibrium can exist only when the two counter input rates are equal and the arriving pulse trains cancel each other. At that point, $FWV = FU$, or $W = U/V$. In the steady state, therefore, the counter holds the desired quotient in coded form. When the quantities U and V are changing, the count W will in general lag behind the true quotient. Analysis of the divider shows that it behaves dynamically like a simple RC network having an effective time constant of $1/VF$. The resulting error in the quotient W depends upon how rapidly U and V vary with respect to this time constant.

In a practical mechanization of the divider, means must be provided to prevent, or interpret properly, the simultaneous arrival at the counter of pulses in the trains FU and FWV. Also, if any of the variables change sign over their range of operation, appropriate switching must be included to sort out the positive and negative pulses.

Another simple illustration of the use of digital-operational techniques is the generation of sine and cosine of a quantity available in incremental form (Fig. 21.52). On the assumption that the multiplication properties of the multipliers hold for the individual input pulses $\Delta\theta_j$ and that counter 2 holds $\cos\theta_i$, multiplier 2 produces the pulse train $\cos\theta_j\,\Delta\theta_j$. Counter 1, by accumulating these pulses, effectively performs an integration:

$$\Sigma \cos\theta_j\,\Delta\theta_j \cong \int \cos\theta\,d\theta = \sin\theta$$

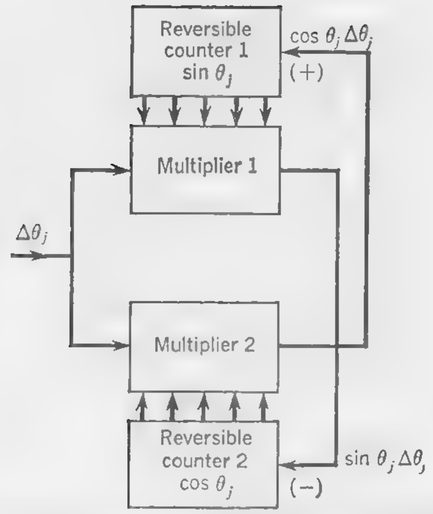


FIG. 21.52. Generation of sine and cosine by digital-operational techniques.

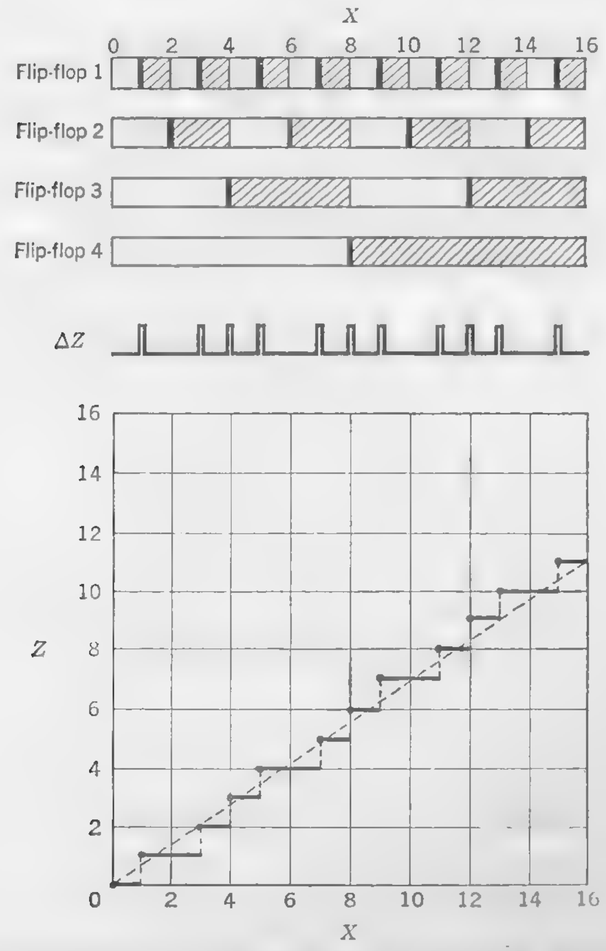


FIG. 21.53. Timing diagram for a four-stage binary operational multiplier. With $Y = 1\frac{1}{16}$, gates are open from stages 1, 2, and 4. Errors of $1\frac{1}{16}$ pulses occur at $X = 13$ and just before $X = 3$.

Similarly, counter 2 receives an input sin $\theta_j \Delta\theta_j$, which is integrated to produce cos θ. The desired trigonometric functions are therefore available as coded numbers in the counter registers. As in the preceding example, sign-control and pulse-separation means have been omitted from the diagram for simplicity.

Accuracy of a Binary Operational Multiplier. The above explanation assumes that a binary operational multiplier at all times maintains the correct multiplicative relationship between its inputs and the output. As pointed out previously, this is not necessarily the case. Figure 21.53 shows a timing diagram for a four-stage multiplier. The states of the individual counter stages are indicated, as well as the noncarry pulses, which occur at the transitions from 0's to 1's. (The diagram also illustrates the fact that no two noncarry pulses can occur at the same time. This is the principle that allows the simple mixing of the gated pulses indicated in Fig. 21.50.) Since the counter cycle repeats every $2^4 = 16$ input pulses, only 16 pulses need be considered. Suppose $Y = 1\frac{1}{16}$ so that 11 Z pulses should be produced for every

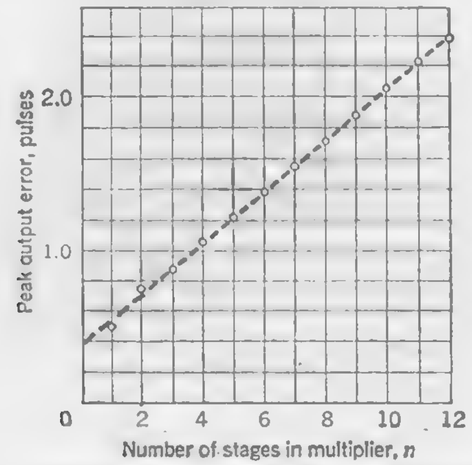


FIG. 21.54. Peak errors produced by an n-stage multiplier over the cycle of 2^n input pulses, assuming Y to be constant.

16 X pulses. In binary notation, $1\frac{1}{16}$ is 0.1011; consequently, gates 1, 3, and 4 are open.

The resulting pulse train, as well as its sum representing Z, are sketched in Fig. 21.53. Considerable deviation between Z and an ideal straight line is evident. After 13 X pulses, for example, 10 Z pulses have been produced, whereas only $1\frac{1}{16} \times 13 = 8\frac{15}{16}$ are called for. The difference between the desired and actual outputs is $1\frac{7}{16}$. If X is regarded as a continuous variable which has been quantized, a similar error of opposite sign occurs just before the third X pulse. At 16 pulses, the exact desired relationship exists, and beyond 16 the cycle repeats; consequently, the errors noted above are of an instantaneous nature. They repeat each cycle of 16 X pulses, but they do not accumulate.

In general, for this mode of operation, the peak error on an n-stage multiplier is

$$E_{\max} = \frac{7}{18} + \frac{n}{6} + \frac{1}{9}(-1)^n 2^{-n} \tag{21.59}$$

For large n, this approaches

$$E_{\max} \cong 0.39 + 0.167n \tag{21.60}$$

A 10-stage counter, for example, produces instantaneous errors as large as 2.1 pulses (see Fig. 21.54). These peak errors occur only for certain combinations of the X and Y inputs. For $n > 2$, they occur twice per cycle at two values of Y—one near $\frac{1}{3}$ and one near $\frac{5}{6}$. Equation (21.60) shows that, although the peak errors increase with

the number of stages, they increase at a relatively slow rate. It is always possible, therefore, to achieve any prescribed precision by decreasing the significance of the least significant digit, even though that means increasing n, the multiplier length.

The preceding discussion applies to the case in which all stages of the multiplier are cleared at $X = 0$ and in which Y remains constant throughout the cycle. When these two conditions do not apply, the errors may be greater and may accumulate from cycle to cycle. Consider, for example, the situation illustrated in Fig. 21.55 in which Y is allowed to change back and forth between $\frac{3}{16}$ and $\frac{4}{16}$ as often as every other X pulse. Ideally, $3\frac{1}{2}$ Z pulses should be produced over the cycle, this being the area under the Y curve. Actually, no Z pulses are produced because no gate is ever

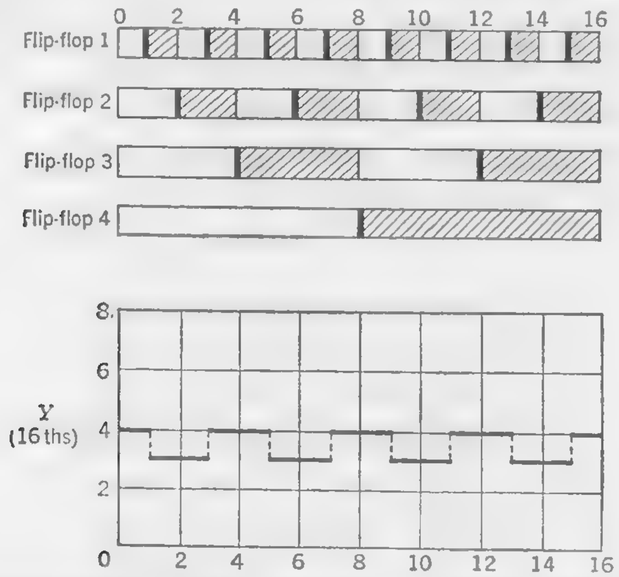


FIG. 21.55. Hypothetical case in which a four-stage multiplier produces an accumulating error of 3.5 pulses for every 16 X pulses.

open when the corresponding noncarry pulse occurs. This error of $3\frac{1}{2}$ then carries over into the next cycle.

Admittedly this is a contrived example. Unless X and Y are otherwise related, it is not likely that Y will change to exactly the right values at exactly the right times to accumulate such an error. Since positive and negative errors are equally likely, depending on the amount of the Y changes and their phasing relative to X, it might be expected that the average error will be small. However, if correlation does exist between X and Y, as in the sine-cosine generation (Fig. 21.52), the type of error indicated by the example of Fig. 21.55 may result. A general treatment of such errors does not appear feasible at the present time.

References

1. Adams, C. W.: Electronic Data Processing in Industry, *Special Report* 3, American Management Association, Inc., New York, 1955.
2. Alderson, Wroe: Operations Research and Management Problems, *Advanced Management*.
3. Amico, G. Vincent: Preface to Space Flight, *ONR Research Rev.*, November, 1955, pp. 15–20.
4. Astrahan, M. M.: The Role of Large Memories in Scientific Communications, *IBM J. Research Development*, **2**(4):310–313 (October, 1958).
5. Baer, J. S., A. S. Rettig, and I. Cohen: On-line Sales Recording System, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 251–256.
6. Baricelli, Nils: Esempi numerici processi di evoluzione, *Methods*, Milan, Italy, **6**(21 22):45–68 (1954).

7. Bauer, Walter F., and George P. West: A System for General-purpose Analog-digital Computation, *J. Assoc. Computing Machinery*, **4**(1):12–17 (January, 1957).
8. Bendz, W. I., and L. J. Kamm: Tape Control Improves Quality Control, *Automatic Control*, **1**(3):10–14 (September, 1954).
9. Bennett, Corwin A.: Some Experimentation on the Tie-in of the Human Operator to the Control Loop of an Airborne Navigational Digital Computer System, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 68–71.
10. Bergen, A. R., and J. R. Ragazzini: Sampled-data Processing Techniques for Feedback Control Systems, *Trans. AIEE*, **73**(part II):236–244 (November, 1954).
11. Binary Digitizer Simplifies Tape Control System, *Design News*, **10**(21):22–23 (Nov. 1, 1955).
12. Bolles, E. E.: The Digitac Airborne Digital Computer, *IRE Convention Record*, part 5, Aeronautical Electronics and Telemetry, New York, 1954, pp. 72–76.
13. Boring Unit . . . Tape Controlled, *Automation*, **2**(9):69–70 (September, 1955).
14. Bracken, R. H., and H. E. Tillitt: Information Searching with the 701 Calculator, *J. Assoc. Computing Machinery*, **4**(2):131–136 (April, 1957).
15. Brooks, F. P., Jr.: A Program-controlled Program Interruption System, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 128–131.
16. Brown, Gordon S.: Trends in Automatization of Procedures and Processes in Business and Industry, *IRE Convention Record*, part 4, New York, 1955, pp. 52–53.
17. Brown, G. W., and L. N. Ridenour: The Processing of Information-containing Documents, *Proc. Western Joint Computer Conf.*, Los Angeles, Feb. 4–6, 1953, pp. 80–85.
18. Brown, J. H.: Measure Motion to 0.0001 Inch without Friction or Wear, *Control Eng.*, **2**(4):50–52 (April, 1955).
19. Burbeck, D. W., E. E. Bolles, W. E. Frady, and E. M. Grabbe: The Digitac Airborne Control System, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S–59, Feb. 11–12, 1954, pp. 38–44.
20. Burns, T. J., J. D. Cloud, and J. M. Salzer: Experiments with a Digital Computer in a Simple Control System, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 60–74.
21. Campbell, Donald P.: Dynamic Behavior of Linear Production Systems, *Mech. Eng.*, **75**(4):279–283 (April, 1953).
22. Canning, R. G.: *Electronic Data Processing for Business and Industry*, John Wiley & Sons, Inc., New York, 1956.
23. Casper, L.: Textile Automation by Signal Control, *Trans. AIEE*, **74**(part I):176–180 (May, 1955).
24. Chung, J. H., and C. C. Gotlieb: Test of an Inventory Control System on FERUT, *J. Assoc. Computing Machinery*, **4**(2):121–130 (April, 1957).
25. Computer Will Control Major Power System, *Control Eng.*, **2**(10):20–21 (October, 1955).
26. Contouring Machine with Tracer Control (Staff Report), *Elec. Mfg.*, **56**(1):135–142 (July, 1955).
27. Control Components Provide New Ways to Handle Materials, *Automatic Control*, **2**(5):27–31 (May, 1955).
28. Control in Two Dimensions, *Automatic Control*, **1**(2):24–25 (August, 1954).
29. Cunningham, F. W.: Controlling Machines With Tape, *Automation*, **1**(1):77–82 (August, 1954).
30. Davis, M. E.: Use of Electronic Data-processing Systems in the Life Insurance Business, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 8–10, 1953, pp. 11–17.
31. DeCarlo, C. R.: Application of Data Processors in Production, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 61–65.
32. Doyle, R. H., R. A. Meyer, and R. P. Pedowitz: Automatic Failure Recovery in a Digital Data Processing System, *IBM J. Research Development*, **3**(1):2–12 (January, 1959).
33. Early, E. D., W. E. Phillips, and W. T. Shreve: An Incremental Cost of Power Delivered Computer, *Trans. AIEE*, **74**(part III):529–534 (June, 1955).
34. Electronic Pantograph in Contouring Machine (Staff Report), *Elec. Mfg.*, **55**(2):88–99 (February, 1955).
35. Electronic Positioning, *Wireless World*, **61**(1):25–27 (January, 1955).
36. Exner, W. L., and A. D. Scarbrough: A Digital Autopilot Coupler, *IRE Convention Record*, part 5, Aeronautical Electronics and Telemetry, New York, 1954, pp. 174–178.
37. Frady, W. E., and M. Phister: System Characteristics of a Computer Controller for Use in the Process Industries, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 40–45.

38. Farrell, Roger H.: Dynamic System Studies: Digital Computers, esp. pp. 1–3, *WADC Tech. Rept. 54–250*, part Y, September, 1956.
39. Fenimore, G. E.: Real-time Data Processing for CAA Air-traffic Control, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 169–172.
40. Forester, J. R., and C. J. Sullivan: Flight Simulation with a Computer, *Automatic Control*, **3**(6):16–19 (December, 1955).
41. Frame, J. S.: The Solution of Equations by Continued Fractions, *Am. Math. Monthly*, **60**:293–305 (May, 1953).
42. Friend, Edward H.: Sorting on Electronic Computer Systems, *J. Assoc. Computing Machinery*, **3**(3):134–168 (July, 1956).
43. Frost, G. R., W. Keister, and A. E. Ritchie: A Throwdown Machine for Telephone Traffic Studies, *Bell System Tech. J.*, **32**:292–359 (March, 1953).
44. Gerlough, D. L.: Control of Automobile Traffic—A Problem in Real-time Computation, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 75–79.
45. Good, I. J.: How Much Science Can You Have at Your Fingertips?, *IBM J. Research Development*, **2**(4):282–288 (October, 1958).
46. Gordon, B. M.: Adapting Digital Techniques for Automatic Controls, *Elec. Mfg.*, **54**(part 1, no. 5):136–143 (November, 1954); (part 2, no. 6):120–125 (December, 1954).
47. Gordon, B. M., and R. N. Nicola: Special Purpose Digital Data-processing Computers, *Proc. Assoc. Computing Machinery*, Pittsburgh, May 2–3, 1952, pp. 33–45.
48. Gordon, B. M.: Application of Operational Digital Techniques to Industrial Control, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, p. 45 (Abstract).
49. Gorn, Saul: The Automatic Analysis and Control of Computing Errors, *J. Soc. for Industrial and Applied Math.*, **2**:69–81 (June, 1954).
50. Gorn, Saul: On the Mechanical Simulation of Habit-forming and Learning, *Information and Control*, **2**(3):226–259 (September, 1959).
51. Grabbe, E. M., D. W. Burbeck, and S. B. Neister: Flight Testing of an Airborne Digital Computer, *IRE Convention Record*, part 5, Aeronautical Electronics and Telemetry, New York, 1954, pp. 66–70.
52. Gray, H. J., Jr.: Numerical Methods in Digital Real-time Simulation, *Quart. Appl. Math.*, **12**:133–140 (July, 1954).
53. Gray, H. J., Jr.: Digital Computer Solution of Differential Equations in Real Time, *Proc. Western Joint Computer Conf.*, Los Angeles, May 6–8, 1958, pp. 87–91.
54. Greenwood, I. A., Jr., J. V. Holdam, Jr., and D. MacRae, Jr., Eds.: *Electronic Instruments*, 21, MIT Rad. Lab. Series, McGraw-Hill Book Company, Inc., New York, 1948.
55. Grieser, Norman: Production Control with the Elecom 125, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 163–171.
56. Gurk, H. M.: The Use of Stability Charts in the Synthesis of Numerical Quadrature Formulae, *Quart. Appl. Math.*, **13**:73–78 (April, 1955).
57. Gurk, H. M., and M. Rubinoff: Numerical Solution of Differential Equations, *Proc. Eastern Joint Computer Conf.*, Philadelphia, Dec. 8–10, 1954, pp. 58–64.
58. Harder, E. L.: Economic Load Dispatching, *Westinghouse Engr.*, **14**(6):191-200 (November, 1954).
59. Hastings, Cecil: *Approximations for Digital Computers*, Princeton University Press, Princeton, N.J. 1955.
60. Hesse, L.: Automatic Turret Lathe Controlled by Magnetic Tape, *Machinery*, **61**(12): 156–160 (August, 1955).
61. Hilderbrandt, Paul, and Harold Isbitz: Radix Exchange—An Internal Sorting Method for Digital Computers, *J. Assoc. Computing Machinery*, **6**(2):156–163 (April, 1959).
62. Ho, Y. C., and E. C. Johnson: Design of a Numerical Milling Machine System, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 11–24.
63. Hobson, J. E.: New Equations for Management, *Proc. Western Joint Computer Conf.*, Los Angeles, Feb. 4–6, 1953, pp. 9–18.
64. Hogan, J. W.: Magnetic Tape Controls Machine Tools, *Electronics*, **27**(12):144–147 (December, 1954).
65. Hornfeck, A. J.: Computing Circuits and Devices for Industrial Process Functions, *Trans. AIEE*, **71**(part I):183–193 (July, 1952).
66. Householder, Alston S.: *Principles of Numerical Analysis*, chaps. 5 and 6; esp. pp. 193–200, 220–225, McGraw-Hill Book Company, Inc., New York, 1953.
67. Huskey, Harry D.: A Solution for Automatic Unit Control, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 96–97.

68. Huskey, H. D., and D. E. Trumbo: Data Preparation for Numerical Control of Machine Tools, 1958 *IRE WESCON Convention Record*, part 4, Aug. 19–22, 1958, pp. 3–7.
69. Irvine, N. L., and L. Davis: Simulation by Modeling, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 13–16.
70. Isaac, E. J., and R. C. Singleton: Sorting by Address Calculation, *J. Assoc. Computing Machinery*, **3**(3):169–174 (July, 1956).
71. Johnson, E. C.: A Numerically Controlled Cam-milling Machine, *Proc. Natl. Industrial Electronics Conf.*, September, 1955.
72. Johnson, Clarence L.: *Analog Computer Techniques*, esp. chap. 13, The Digital Integrating Differential Analyzer, McGraw-Hill Book Company, Inc., New York, 1956.
73. Johnson, D. L.: Generating and Testing Pseudo-random Numbers on the IBM Type 701, *Mathematical Tables and Other Aids to Computation*, **10**:8–13 (January, 1956).
74. Juncosa, M. L.: Random Number Generation on the BRL High-speed Computing Machines, Ballistic Research Laboratories, Aberdeen Proving Ground, Maryland, *Rept. 855*, 1953.
75. Kelling, L. U. C.: Numerical Control of Machine Tools, *IRE Trans. on Industrial Electronics*, **2**:3–8 (March, 1955).
76. Kohr, Robert H.: Application of Computers to Automobile Control and Stability Problems, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 84–89.
77. Leaver, E. W., and G. R. Mounce: Recorder-controlled Automatic Machine Tools, *Electronics*, **27**(11):124–128 (November, 1954).
78. Linvill, W. K., and J. H. Salzer: Analysis of Control Systems Involving Digital Computers, *Proc. IRE*, **41**:901–906 (July, 1953).
79. Linvill, W. K.: Sampled-data Control Systems Studied through Comparison of Sampling with Amplitude Modulation, *Trans. AIEE*, **70**(part II):1779–1788 (1951).
80. Lotkin, Mark: On the Improvement of Accuracy in Integration, *Quart. Appl. Math.*, **13**:47–54 (April, 1955).
81. Luhn, H. P.: A Business Intelligence System, *IBM J. Research Development*, **2**(4): 314–319 (October, 1958).
82. Luhn, H. P.: The Automatic Creation of Literature Abstracts, *IBM J. Research Development*, **2**(2):159–165 (April, 1958).
83. Malthaner, W. A., and H. E. Vaughn: An Automatic Telephone System Employing Magnetic Drum Memory, *Proc. IRE*, **41**(10):1341–1347 (October, 1953).
84. Margolis, M., and E. Weiss: Simulation Techniques Associated with a Digital Flight Control System, *Proc. Symposium III on Simulation and Computing Techniques*, October, 1953, pp. 306–324.
85. Margolis, M., and Eric Weiss: An Experimental Digital Flight Control System, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 23–37.
86. Mason, Robert M.: The Digital Approximation of Contours, *J. Assoc. Computing Machinery*, **3**(4):355–359 (October, 1956).
87. McAvoy, R. A.: Reservations Communications Utilizing a General Purpose Digital Computer, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 178–182.
88. McCloskey, J. F., and F. N. Trefethen, Eds.: *Operations Research for Management*, Johns Hopkins Press, Baltimore, 1954.
89. McDonald, John: *Strategy in Poker, Business and War*, W. W. Norton & Company, Inc., New York, 1950.
90. McDonough, J. O., and A. W. Susskind: A Numerically Controlled Milling Machine, *Review of Input and Output Equipment Used in Computing Systems, Joint AIEE-IRE-ACM Computer Conf.*, New York, S-53, Dec. 10–12, 1952, pp. 133–136.
91. McFerren, E. L.: Control Tape Prepared from Numerical Data, *Machinery*, **61**(12): 178–182 (August, 1955).
92. McGalliard, David H.: Business Judgment Stimulated through Management Simulation, *Univac Rev.*, **2**(1):20–21 (winter, 1959).
93. McPherson, J. L.: Commercial Applications—The Implication of Census Experience, *Proc. Western Joint Computer Conf.*, Los Angeles, Feb. 4–6, 1953, pp. 49–53.
94. Meissner, L. P.: Real-time Digital Differential Analyzer, *Proc. Western Joint Computer Conf., Trends in Computers: Automatic Control and Data Processing*, Los Angeles, Feb. 11–12, 1954, pp. 131–139.
95. Mergler, H. W.: A Digital-analog Machine Tool Control System, *Proc. Western Joint Computer Conf., Trends in Computers: Automatic Control and Data Processing*, Los Angeles, Feb. 11–12, 1954, pp. 46–59.

96. Mergler, H. W., G. J. Moshos, and A. E. Young: Machine Tool Control from a Digital-analog Computer, *IRE Trans. on Industrial Electronics*, **1**:26–30 (August, 1953).
97. Meyer, M. A.: Digital Techniques in Analog Systems, *IRE Trans. on Electronic Computers*, **EC-3**(2):23–29 (June, 1954).
98. Meyer, M. A., B. M. Gordon, and R. N. Nicola: An Operational-digital Feedback Divider, *IRE Trans. on Electronic Computers*, **EC-3**(1):17–20 (March, 1954).
99. Milne-Thomson, L. M.: *The Calculus of Finite Differences*, Macmillan & Co., Ltd., London, 1933.
100. Milne, William E.: *Numerical Solution of Differential Equations*, John Wiley & Sons, Inc., New York, 1953.
101. Moore, Gerald T.: The Numericord Machine-tool Director, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 6–10.
102. Morgan, M.: Punched Tape Control of Cam-milling Machine, *Elec. Mfg.*, **56**(4): 114–125 (October, 1955).
103. Morrill, C. D., and J. A. Blake: A Computer for Economic Scheduling and Control of Power Systems, *Trans. AIEE*, **74**(part III):1136–1141 (December, 1955).
104. Moshos, G. J.: Analog Interpolator for Automatic Control, *J. Assoc. Computing Machinery*, **2**(2):83–91 (April, 1955).
105. Muller, Mervin E.: The Use of Computers in Inspection Procedures, *Commun. Assoc. Computing Machinery*, **1**(11):7–13 (November, 1958).
106. National Bureau of Standards, Applied Mathematics Series, 9, *Tables of Chebyshev Polynomials* $S_n(x)$ and $C_n(x)$, Washington, D.C., 1952.
107. Newell, A.: The Chess Machine: An Example of Dealing with a Complex Task by Adaptation, *Proc. Western Joint Computer Conf.*, Los Angeles, Mar. 1–3, 1955, pp. 101–108.
108. Nichols, J. R.: Tape Gives Turret Lathe Short-run Flexibility, *Am. Machinist*, **99**(16): 106–109 (Aug. 1, 1955).
109. Norden, P. V.: Curve Fitting for a Model of Applied Research and Development Scheduling, *IBM J. Research Development*, **2**(3):232–248 (July, 1958).
110. Nuclear Codes Group, Virginia Nather and Ward Sangren, Abstracts—Nuclear Reactor Codes, *Commun. Assoc. Computing Machinery*, **2**(1):6–32 (January, 1959).
111. Oettinger, Anthony G.: Account Identification for Automatic Data Processing, *J. Assoc. Computing Machinery*, **4**(3):245–253 (July, 1957).
112. Otis, E. J.: Optimized Control through Digital Equipment, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 45–49.
113. Payne, A. H.: Stock Transaction Records on the Datatron 205, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 183–186.
114. Peaslee, L. R.: Tape-controlled Machines, *Elec. Mfg.*, **52**(5):102–108 (November, 1953).
115. Proceedings on the Conference on What Is Operations Research Accomplishing in Industry? Case Institute of Technology, Cleveland, Ohio, 1955. Also proceedings of similar conferences, 1951, 1953, and 1954.
116. Ragazzini, J. R., and L. A. Zadeh: The Analysis of Sampled Data Systems, *Trans. AIEE*, **71**(part II):225–234 (November, 1952).
117. Recorded Blueprint Dimensions Control Machine Tools, *Automatic Control*, **3**(1): 26–27 (July, 1955).
118. Retzinger, L. P., Jr.: An Input-output System for a Digital Control Computer, *Proc. WESCON Computer Sessions*, Los Angeles, Aug. 25–27, 1954, pp. 67–76.
119. Rotkin, I.: The Mechanization of Letter Mail Sorting, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 54–57.
120. Rubinoff, Morris: Analogue vs. Digital Computers—A Comparison, *Proc. IRE*, **41**:1254–1262 (October, 1953).
121. Rubinoff, Morris: Digital Computers for Real-time Simulation, *J. Assoc. Computing Machinery*, **2**(3):186–204 (July, 1955).
122. Rubinoff, Morris: Some Recent Developments in Real-time Digital Simulation and Control, *Proc. First Flight Simulation Symposium*, White Sands Proving Ground, White Sands, N.M., September, 1957, pp. 20–33.
123. Rutishauser, von Heinz: Über die Instabilität von Methoden zur Integration gewohnlicher Differentialgleichungen, *Z. angew. Math. Phys.*, **3**:65–74 (1952).
124. Salveson, M. E., and R. G. Canning: Automatic Data Processing in Larger Manufacturing Plants, *Proc. Western Joint Computer Conf.*, Los Angeles, Feb. 4–6, 1953, pp. 65–73.
125. Salzer, John M.: System Compensation with a Digital Computer, *IRE Convention Record*, part 5, Aeronautical Electronics and Telemetry, New York, 1954, pp. 179–186.

126. Scarborough, James B.: *Numerical Mathematical Analysis*, 2d ed., esp. chaps. 1–6, 16, Johns Hopkins Press, Baltimore, 1950.
127. Scarbrough, A. D.: An Analog-to-digital Converter, *IRE Trans. on Electronic Computers*, **EC-2**(3):5–7 (September, 1953).
128. Schelkunoff, Sergei A.: *Applied Mathematics for Engineers and Scientists*, esp. pp. 36–38, D. Van Nostrand Company, Inc., Princeton, N.J., 1948.
129. Schneider, D. B., and H. V. Nelson: Numerical Positioning Control (Staff Report), *Elec. Mfg.*, **56**(3):122–127 (March, 1955).
130. Selfridge, R. G.: Approximations with Least Maximum Error, *Pacific J. Math.*, **3**:247–255 (March, 1953).
131. Shaffer, S. J.: Ready-to-wear Unit Control Procedure, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 82–88.
132. Shannon, C. E.: Computers and Automata, *Proc. IRE*, **41**:1234–1241 (October, 1953).
133. Shannon, C. E., and E. F. Moore: Machine Aid for Switching Circuit Design, *Proc. IRE*, **41**:1348–1351 (October, 1953).
134. Sopka, J. J.: An Analysis of Adequate Inventory Levels, *IBM J. Research Development*, **3**(1): 54–57 (January, 1959).
135. Stoker, J. J.: On the Stability of Mechanical Systems, *Commun. Pure Appl. Math.*, **8**:133–141 (February, 1955).
136. Stone, J. J., Jr., B. B. Gordon, and R. S. Boyd: Physical Simulation of Nuclear Reactor Power Plant Systems, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 80–83.
137. Tape-controlled Circuit Wiring, *Automatic Control*, **3**(4):8–9 (October, 1955).
138. Taylor, C. H., Jr.: A Coordinated Data-processing System and Analog Computer to Determine Refinery-process Operating Guides, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 34–39.
139. Teichman, T.: Closed Loop Control Systems Containing a Digital Computer, *IRE Trans. on Electronic Computers*, **EC-4**(3):106–117 (September, 1955).
140. Teichroew, D.: Use of Continued Fractions in High Speed Computing, *Mathematical Tables and Other Aids to Computation*, **6**:127–132 (July, 1952).
141. Thomas, A. G.: Electronic Tape Control System Uses Step Motor Drive, *Automation*, **2**(3):62–63 (March, 1955).
142. Tou, Julius: High Accuracy Operational Digital Simulation, *Proc. Natl. Simulation Conf.*, January, 1956, pp. 11.1–11.6.
143. Truxal, John G.: *Automatic Feedback Control System Synthesis*, chap. 9, McGraw-Hill Book Company, Inc., New York, 1955.
144. Tustin, A. (Ed.): *Automatic and Manual Control*, esp. pp. 13–44, Butterworth & Co. (Publishers), Ltd., London, 1952.
145. Wall, H. S.: *Analytic Theory of Continued Fractions*, D. Van Nostrand Company, Inc., Princeton, N.J., 1948.
146. Wedel, J. J., A. Huntington, and M. B. Bain: Automatic Data-accumulation System for Wind Tunnels, *IRE Trans. on Electronic Computers*, **EC-5**(1):7–11 (March, 1956).
147. Whitby, Oliver: The Automatic Handling of Business Data, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 75–79.
148. Wilks, P. A., Jr.: Analysis Moves from Lab to Line, *Control Eng.*, **1**(2):42–46 (October, 1954).
149. Williamson, D. T. N.: Computer Controlled Machine Tools, *Trans. AIEE*, **74**(part I): 70–74 (August, 1955).
150. Winter, A. J.: A Magnetically Coupled Low-cost High-speed Shaft Position Digitizer, *Proc. Western Joint Computer Conf.*, Los Angeles, Feb. 4–6, 1953, pp. 203–207.
151. Wong, S. Y.: Traffic Simulator with a Digital Computer, *Proc. Western Joint Computer Conf.*, San Francisco, Feb. 7–9, 1956, pp. 92–94.
152. Wooldridge, Dean E.: Trends in Electronic Business Data Systems Development, *Trends in Computers: Automatic Control and Data Processing, Proc. Western Joint Computer Conf.*, Los Angeles, S-59, Feb. 11–12, 1954, pp. 16–22.
153. Zadoff, S., and J. Rattner: Use of a Digital Computer for Airborne Guidance and Navigation, *Proc. Eastern Joint Computer Conf.*, Washington, D.C., Dec. 9–13, 1957, pp. 64–68.

INDEX

Abbreviations for analog computers, **1**-6
Absolute-value circuit, **3**-73
A-C amplifiers, resistance-coupled, **2**-47, **2**-48
A-C analog computers, **8**-16 to **8**-18
A-C network analyzer, **9**-61 to **9**-68
A-C synchronous motors, **18**-47, **18**-48
Acceleration error, **5**-37, **5**-39
Acceleration limit, **3**-19
Accelerometer, simulation of, **5**-35
Access time, **12**-32, **12**-33, **12**-90, **12**-117
  information transfer and, **16**-15, **16**-16
Accumulator, carry operation, **16**-25 to **16**-27
  examples of, **20**-11, **20**-13, **20**-17
  parallel binary, **15**-17
  serial binary, **15**-12, **15**-13, **16**-13
  serial register and, **15**-10, **15**-11
Add-multiply command, **17**-24
Add-subtract, selective (binary), **18**-33, **18**-34
Add-subtract counters, **18**-34
Adder, circuit design, **14**-18
  digital differential analyzer, **19**-39, **19**-40
  root-mean-square, **3**-76, **7**-25, **7**-26, **8**-21
  serial binary, **15**-9, **15**-10
  serial half-adder, **15**-8, **15**-9
Adder-subtractor, mechanical analog, **8**-2 to **8**-4
Addition, analog, basic circuits, **5**-5 to **5**-7
  mechanical, **8**-2 to **8**-4
  of RMS values, **8**-21
  digital, binary, **15**-2
  decimal, **15**-21
  floating-point numbers, **15**-22, **15**-23
  octal, **15**-3
  digital differential analyzer, **19**-39 to **19**-42
Address, digital differential analyzer, **19**-25, **19**-71
  IBM 650, **20**-10
  LGP-30, **20**-15
  sequence, **16**-5
  stored-program computer systems, **16**-8 to **16**-11
Address modification, **20**-28 to **20**-30
Adjoint-system method, **5**-101, **5**-107
Administration, analog computers, large installations, **4**-41 to **4**-44
  (*See also* Management problems)
Aerodynamic coefficients, **5**-51
Aeroelasticity, **5**-62
AGC for resolvers, **3**-27 to **3**-29
Aircraft, control systems, analog, **5**-60 to **5**-62
  flight equations, **5**-49 to **5**-60
  flight simulation, analog, **5**-64
  digital, **21**-56 to **21**-59
  flight tables, **5**-65 to **5**-71
  navigation systems, digital, **21**-81 to **21**-84
Aircraft Marine Products, AM-P shielded patchbay system, **4**-27
Airfoil mapping, **6**-56
Algebra, Boolean, **14**-14, **16**-12
  (*See also under* Boolean)
Algebraic equations (*see* Equation solving)
ALGOL 60, **17**-4, **17**-26 to **17**-37, **20**-20
  examples, **17**-29 to **17**-31, **17**-37
  syntax, of declarations, **17**-35, **17**-36
  of expressions, **17**-31 to **17**-34
  of statements, **17**-34, **17**-35
Aliasing errors, **6**-26, **6**-43 to **6**-48
Alphabets, **14**-7 to **14**-12
  group, **14**-7, **14**-8, **14**-10, **14**-11
  Hamming, **14**-8 to **14**-10
  parity-check, **14**-7, **14**-8
  Plotkin, **14**-12
  Reed-Muller, **14**-11, **14**-12
  (*See also* Symbols, coding)
AM-AM multiplier, **3**-46, **3**-47
AM-FM multiplier, **3**-48, **3**-52
AM-PM multiplier, **3**-48 to **3**-50, **7**-12 to **7**-14
Ampere, **10**-4
Ampex FR 1100 tape transport, **6**-12, **6**-13
Amplifiers, analog computer, a-c resistance-coupled, **2**-47, **2**-48
  circuits, **2**-17 to **2**-20, **2**-72 to **2**-82
  test, **4**-44 to **4**-46
  d-c (*see* D-C amplifiers)
  direct-analogy network, **9**-45 to **9**-48
  errors in, **2**-8, **2**-41, **2**-57, **2**-58, **7**-29
  parallel-feedback, **2**-8, **2**-16, **2**-17
  performance equations, **2**-8, **2**-41, **2**-42
  repetitive computers, **6**-20 to **6**-22
  specifications, **2**-73
  transfer impedances, **2**-20 to **2**-27
  digital computer, **11**-13 to **11**-38
  amplifier response, **11**-1, **11**-2
  audio, *RC* coupled, **11**-25 to **11**-28
  transformer-coupled, **11**-33, **11**-34
  cathode-coupled, **11**-24
  direct-coupled, **11**-18 to **11**-24
  gating, **12**-125
  read, **12**-88 to **12**-90

Amplifiers, digital computer, regulator, **11**-24, **11**-25
tuned, **11**-34 to **11**-36
video, **11**-28 to **11**-33
Williams-tube system, **12**-39, **12**-40
Amplitude-distribution analyzers, **5**-106 to **5**-109, **6**-19, **6**-31, **6**-32
Amplitude modulation, tape recording, **6**-10
Amplitude scaling, **5**-12, **5**-13
Amplitude-selection multiplier (*see* Triangle-integration multiplier)
Amplitude selector, **3**-70, **3**-80
Analog comparator (*see* Comparator)
Analog computers, abbreviations, **1**-6
a-c/electromechanical, **8**-16 to **8**-18
applications, **1**-5
table, **5**-23 to **5**-34
(*See also* Programming; specific applications)
block-diagram symbols, **1**-6 to **1**-10, **5**-5, **8**-4
combined with digital computers, **6**-42 to **6**-52
compared with digital computers, **1**-4, **1**-5
as control-system components, **6**-35 to **6**-42
d-c, **1**-4
compared with a-c, **8**-17, **8**-18
definition, **1**-3, **1**-4
direct-analogy, **9**-3 to **9**-87
large installations, **4**-40 to **4**-44
types, comparative data, **4**-43
mechanical, **8**-2 to **8**-16
network analyzers, **9**-61 to **9**-68
repetitive, **6**-17 to **6**-25, **6**-31 to **6**-34
small and medium-sized, **4**-34 to **4**-39
desk-top, **4**-38, **4**-39
expandable, **4**-39
transistorized, **7**-1 to **7**-36
Analog controller, **6**-40
Analog-digital computers (*see* Hybrid analog-digital computers)
Analog-to-digital converters, **3**-71, **6**-44, **6**-45, **18**-29 to **18**-42
binary-increment multiplier, **7**-15 to **7**-18
with digital differential analyzers, **19**-55
frequency-to-number, **18**-41, **18**-42
position-to-number, **18**-35 to **18**-41
time-to-number, **18**-41
voltage-to-number, **18**-30 to **18**-35
Analog-digital simulation, combined, **6**-42 to **6**-52, **21**-54 to **21**-56
Analog memory pair, **6**-32 to **6**-34, **6**-54
Analogous quantities, **9**-3 to **9**-6
AND, logical operator, **17**-33
AND circuit, logical, **14**-13, **16**-12
negative, **13**-1, **13**-2, **13**-4
positive, **13**-1, **13**-2, **13**-4
AND gate, **13**-1 to **13**-6, **16**-19
crystal-diode, **13**-4, **13**-5
notation, **15**-5
parametric amplifier, **13**-14
Angle of attack, **5**-53, **5**-54
Approximations of time delay, higher-order, **6**-5, **6**-7
lumped-parameter, **6**-5, **6**-6
Arccosine generator, **7**-22, **7**-23
Arcsine generator, **7**-21 to **7**-23
Arithmetic, **15**-1 to **15**-26
binary (*see* Binary numbers)
decimal, **15**-21, **20**-7
digital computer systems, classification by, **20**-6, **20**-7
parallel, **15**-17 to **15**-21, **16**-20 to **16**-27, **19**-26
in relay-type computers, **20**-4, **20**-5
sequence scheme, **15**-23, **20**-23
serial, **15**-8 to **15**-16, **16**-18 to **16**-20, **20**-11, **20**-14, **20**-17
serial-parallel, **15**-19 to **15**-21
Arithmetic channels, digital differential analyzer, **19**-19 to **19**-22
Arithmetic command list, **17**-12 to **17**-14
Arithmetic expressions, ALGOL 60, **17**-32
Arithmetic units, **15**-10 to **15**-26
digital differential analyzer, **19**-26
parallel, **15**-17 to **15**-19
serial, **15**-8 to **15**-16, **16**-18 to **16**-20, **20**-11, **20**-14, **20**-17
serial-parallel, **15**-19 to **15**-21
structure, **15**-24 to **15**-26
timing and synchronization, **16**-17, **16**-18
parallel, **16**-20, **16**-21
serial, **16**-18 to **16**-20
Arizona, University of, simulation board, **4**-37
Arrays, **17**-5, **17**-27, **17**-36
Arrhenius equation, **5**-83
Asymmetrical networks, **9**-76
Attenuation, **11**-8
Autocorrelation function, **5**-94, **5**-95
analog-computer methods for finding, **5**-98 to **5**-101
measurement of (*see* Correlators)
Autocycler, **4**-12, **4**-13
Automatic gain control for resolvers, **3**-27 to **3**-29
Automatic programming, analog computers, **6**-30 to **6**-34, **9**-67
digital computers, **17**-25, **17**-26, **21**-41
Automatic sequential computation, **6**-32 to **6**-34

Backlash, in computer servos, **3**-13, **3**-16
simulation of, **5**-43, **5**-45 to **5**-49
BALANCE CHECK mode, **4**-8
Balancing, automatic, of d-c amplifiers, **2**-51 to **2**-56
Ball-and-disk integrator, **8**-4
Balser-Silverman-Wagner channel and detector, **14**-12, **14**-13
Bank check reading, **18**-19, **18**-20
Barium titanate, **12**-107
Base-case solution, **9**-67
Base quantities for network analyzers, **9**-63

Beam equations, **5**-110
analog solution, **5**-116 to **5**-119, **5**-129 to **5**-132
direct-analogy computers, **9**-12 to **9**-19, **9**-37
Beam power tubes, **10**-14
Beam rider, **5**-60, **5**-61
Beckman/Berkeley, Model 1048B, **2**-68, **2**-73, **2**-74, 3-59
Model 1148, **2**-73, **2**-74
Bell function, **5**-18, **5**-19
Bell-Strauss pile simulator, **5**-88
Bell Telephone Laboratories, DIAD system, **19**-13
relay machine, **19**-9
Bendix flight table, **5**-69
Bendix G-15 computer, **20**-15 to **20**-19
Berkeley Computation Center, Los Angeles, **4**-43
Bidirectional limiters, transistor, **7**-20, **7**-21
Biharmonic equation, analog solution, **9**-79, **9**-80
(*See also* Beam equations)
Binary accumulator, parallel, **15**-17
serial, **15**-12, **15**-13, **16**-13
Binary adder, serial, **15**-9, **15**-10
Binary addition, **15**-2
Binary-arithmetic units, parallel, **15**-17 to **15**-19
serial-parallel, **15**-19 to **15**-21
Binary-coded-decimal attenuator, **7**-16
Binary coding (*see* Coding)
Binary counters, **15**-7, **21**-90
Binary division, serial, **15**-15, **15**-16
Binary-increment multiplier, **7**-15 to **7**-18
Binary multiplication, **15**-2
serial, **15**-13, **15**-14
stored-carry, **15**-18, **15**-19
Binary numbers, **14**-1, **14**-2, **15**-2, **15**-3
coded decimals, **14**-2, **14**-3, **15**-4, **15**-5
reflected, **18**-40
representation of integers, **15**-2
Binary operational multiplier, **21**-74, **21**-75, **21**-89 to **21**-94
Binary weighted-discharge circuit, **18**-43
Biquinary code, **14**-4, **15**-4
Bistable condenser, **12**-110 to **12**-112
Bistable magnetic cores (*see* Magnetic cores)
Block diagrams and symbols for analog computers, **1**-6 to **1**-10
dynamical-system studies, **5**-42 to **5**-49
mechanical, **8**-4 to **8**-12
programming, **5**-5 to **5**-12
servomechanisms, **1**-9, **1**-10, **5**-36 to **5**-42
Blocks (compound statements), **17**-28
Bode plots, differential-analyzer representation of transfer functions, **2**-28 to **2**-40
Body axes, **5**-50
Body Coordinate System (BCS), **5**-50
Boolean algebra, **14**-14, **16**-12
Boolean expressions, ALGOL 60, **17**-32, **17**-33
in Polish notation, **19**-7
Boolean expressions, vector, **17**-5
Boolean logic, origin of, **19**-5
Boolean numbers, **17**-4, **17**-5
Boolean variables, **17**-2
Bootstrap circuit, **2**-81, **2**-82
Bottle plug, **4**-30, **4**-36
Boundary conditions for beam problems, **9**-14, **9**-15
Boundary-value problems, analog solution, **5**-112 to **5**-119
digital solution, **21**-6, **21**-7
Bounded-distance detectors, **14**-12
Bridge limiter, **3**-66
block-diagram symbol, **1**-10
Bucket-brigade approximation of time delay, **6**-8
Buffer inverters, **11**-5 to **11**-7, **15**-5, **16**-12
Buffer storage, **12**-92, **16**-5, **20**-7, **20**-8
Buffer storage unit, **12**-91 to **12**-97
in analog-to-digital conversion, **6**-44, **6**-45
Burroughs computer using Polish notation, **19**-7, **19**-8
Burroughs electrostatic printer, **18**-7
Bush analyzer, **19**-14
Business, data processing, **21**-13 to **21**-19

Cabinets, analog computers, **4**-33, **4**-34
Cabling capacitances, **4**-32, **4**-33
Calculus, propositional, **14**-13, **14**-14
California, University of, digital computer (CALDIC), **12**-32
California Institute of Technology, polynomial root extractor, **19**-9
pulse-height sorter, **19**-12
Cam-milling machine, **21**-75 to **21**-77
Cam potentiometer, **3**-33
Cams, **8**-2, **8**-6, **8**-7
Canadian Westinghouse Co., Ltd., **6**-8, **6**-10, **6**-11
Cap and cup, **14**-14
Capacitances, stray, in analog computers, **4**-32, **4**-33
Capacitor discharge to least significant digit, **18**-35
Capacitor-storage time delay, **6**-8 to **6**-10
Capacitor wheel, **6**-8, **6**-10, **6**-11
Capacitors for analog computers, **2**-55, **2**-60 to **2**-64
equivalent circuits, **2**-61, **2**-62
network type, **9**-37, **9**-39, **9**-44, **9**-45, **9**-62
Carbon paper, **18**-51
Cards, magnetic, **18**-19
punched, **18**-8 to **18**-13
alphanumeric, **18**-8, **18**-9
binary, **18**-8, **18**-9
converters to magnetic tape, **18**-11
programming, **16**-5
punching, **18**-12, **18**-13
reading, **18**-8, **18**-10 to **18**-12
sorting (*see* Sorting)
trailer, **21**-10 to **21**-21

Carry operation, arithmetic, **15**-8 to **15**-10, **15**-18, **16**-25 to **16**-27
Carry suppression, **15**-10
Cathode-bias resistor, effect on d-c amplifier, **2**-45, **2**-46
Cathode-coupled amplifiers, **11**-24
Cathode-follower coincidence circuit, **13**-2, **13**-3
Cathode followers, **11**-2 to **11**-4, **11**-22 to **11**-24, **13**-2
Cathode-ray-tube function generator, **3**-68 to **3**-70, **3**-73, **3**-74, **3**-79, 3-80
Cathode-ray-tube scanners, **3**-69, **3**-70
Cathode-ray tubes, in analog-to-digital converters, **18**-32, **18**-39
  recording techniques, **18**-57, **18**-58
  symbol displays, **18**-25 to **18**-29
  Williams-tube system, **12**-40, **12**-41
Characters (*see* Alphabets; Symbols)
Charactron printer, **18**-7
Chebyshev polynomial, **21**-47 to **21**-49
Check reading, **18**-19, **18**-20
Checking analog computers, **5**-19 to **5**-22
  network type, **9**-59
  (*See also* Problem checking)
Chemical reactions, simulation of, **5**-82 to **5**-85
Chopper, electromechanical, **7**-3, **7**-4
Chopper amplifier, transistor, **7**-32 to **7**-34
Chopper checker, **4**-55
Chopper stabilization of d-c amplifier, **2**-51 to **2**-55, **7**-2 to **7**-6, **7**-32 to **7**-34
Circuits, analog computer, amplifier and power-supply, examples, **2**-72 to **2**-82
  differential-analyzer representation of transfer functions, **2**-28 to **2**-40
  diode limiter (*see* Limiter circuits)
  for dynamical-system studies, **5**-42 to **5**-49
  programming, basic, **5**-5 to **5**-12
  test, **4**-44 to **4**-54
  digital computer, AND (*see* AND circuit)
  carry, **14**-17, **15**-9
  clipping, **11**-8, **11**-32
  coupling, **10**-7, **11**-34
  delay, **11**-9, **11**-10, **16**-12
  end-pulse, **11**-13
  logic in design, **14**-17, **14**-18
  logical, elementary, **16**-11, **16**-12
  microwave, **11**-37
  OR (*see* OR circuit)
  parallel, **10**-6
  pulse-broadening, **11**-11 to **11**-13
  pulse-generating, **11**-13
  RC, **10**-6, **10**-7
  RLC, **10**-8
  series, **10**-6
  signal-modifying, **11**-7 to **11**-13
  single-input component, **11**-1 to **11**-38
  sum, **14**-17
  switching, **13**-1 to **13**-16
  magnetic-core, **12**-64 to **12**-90
Clamp circuit, **11**-5
Clear-and-add, **15**-12
Clear-and-subtract, **15**-12
Clipping circuits, **11**-8, **11**-32
Clock, constant-frequency, **16**-18
  word, **16**-14
Clock pulse, **15**-6, **16**-19
  time-to-number converters, **18**-41, **18**-42
Closed-shop operation of computers, **4**-42
Clutch, simulation of, **5**-41
COBOL, **21**-41, **21**-42
Code wheel, position-to-number converter, **18**-36, **18**-39, **18**-40
Code-wheel servo, **18**-45, **18**-46, **18**-48
Coded-disk digitizer, **21**-67, **21**-68
Coding, **14**-1 to **14**-13, **17**-2 to **17**-38
  alphabets (*see* Alphabets)
  binary, **14**-1, **14**-2, **15**-2
  decimal, **14**-2, **14**-3, **15**-4, **15**-5
  reflected, **18**-40
  biquinary, **14**-4, **15**-4
  cyclic, binary, **18**-40, **18**-41
  for scalar product, **17**-8
  defined, **17**-2
  error detection, **14**-4, **14**-6, **14**-7
  geometrical models, **14**-5, **14**-6
  items, **17**-4
  methods compared, **17**-19, **17**-20
  redundancy, **14**-3 to **14**-5
  relay-type digital computers, **20**-3
  symbols (*see* Symbols)
  teletype, **18**-2, **18**-3
  (*See also* Programming)
Coefficient-setting potentiometer (*see* Potentiometers)
Coincidence circuit, **13**-1 to **13**-6, **15**-5, **16**-12
  cathode-follower, **13**-2, **13**-3
  crystal-diode, **13**-4
  vacuum-diode, **13**-1, **13**-2
  vacuum-tube, **13**-3
Coincidence gate with memory, **13**-6, **13**-7
Coincident-current selection, **12**-52 to **12**-55
  matrices, **12**-53 to **12**-56
Coincident-current storage units, **12**-80 to **12**-86
  driving and switching systems, **12**-81 to **12**-86
  maintenance, **12**-102, **12**-103
  system design, **12**-98 to **12**-100
Closed-loop gain, measurement, **4**-45
Command list, **17**-10, **17**-12 to **17**-14
Command-pulse distributor, **16**-31, **16**-32
Command system, **17**-23, **17**-24
  four-address, **17**-24
  single-address, **17**-23
  three-address, **17**-23
  two-address, **17**-23
Commands, add-multiply, **17**-24
  analog, conversion to, **21**-68, **21**-69
  arithmetic, **17**-12, **17**-13
  Bendix G-15, **20**-17
  IBM 650, **20**-10 to **20**-12
  input and output, **17**-13, **17**-14
  merge, **17**-25
  normalize, **17**-25
  program-logic, **17**-14, **17**-15

Commands, repeat, **17**-24
  Royal-McBee LGP-30, **20**-13
  sequencing of, **16**-28 to **16**-32, **17**-10, **20**-7
  four-address computer, **20**-26, **20**-27
  one-address computer, **20**-21, **20**-23, **20**-24
  with address modification, **20**-28
  two-address computer, **20**-25
  shift-jump, **17**-24, **17**-25
  sift, **17**-25
  summarize, **17**-25
Commutator stabilization, **2**-55, **2**-56
Comparator, analog-voltage, **3**-63, **3**-70, **5**-106, **5**-107
  in analog-to-digital converter, **7**-17
  in analog memory, **6**-32 to **6**-34
  relay, **3**-63
  applications, **5**-40, **5**-46, **5**-48, **5**-153
  block-diagram symbol, **1**-10
Comparison potentiometer, **2**-5, **2**-6
Compilers, **17**-25
Complement, **14**-3, **15**-11, **15**-14, **15**-15
Complementary-symmetry stage, **7**-4
Complete decoding nets, **14**-20, **14**-21
COMPUTE mode, **4**-4
  (*See also* Modes)
Computer Systems, Inc., **3**-52, **4**-43
  CSI 5900, **4**-29
  DYSTAC, **6**-32 to **6**-34
  Model MC-508-C, **2**-73, **2**-76
Computers (*see* Analog computers; Digital computers; specific models)
Condenser, **10**-5
  bistable, **12**-110 to **12**-112
Condenser memory, **12**-116 to **12**-126
  access drivers, **12**-124
  block diagram, **12**-124
  holding equations, **12**-120 to **12**-122
  reading equations, **12**-123, **12**-124
  writing equations, **12**-122, **12**-123
Conditional expressions, ALGOL 60, **17**-33, **17**-34
Conductive-liquid analogies, **9**-82 to **9**-84
Conductive rubber, **9**-81
Conductive-solid analogies, **9**-80 to **9**-82
Cone integrator, **8**-5
Conformal mapping, analog method, **6**-56
Conformity of potentiometers, **3**-5 to **3**-9
Connectives, logical, **14**-13
Constraints to increase accuracy (*see* Steepest-descent methods)
Continuous resolver, **3**-29 to **3**-33, **3**-61, **3**-73
Contouring, **21**-79 to **21**-81
Contradiction, **14**-15
Control, management, **21**-33, **21**-37
Control applications, digital computers, **21**-62 to **21**-94
  aircraft navigation, **21**-81 to **21**-84
  continuous-process control, **21**-84 to **21**-86
  electric-power generation, **21**-86 to **21**-89
  equation solving, **21**-81
  machine-tool control, **21**-63 to **21**-81
Control field, **3**-11
Control panel for analog computer, **4**-9, **4**-10, **9**-58
Control systems, analog computers, **4**-3 to **4**-12, **6**-35 to **6**-42
  circuits, **4**-3 to **4**-9
  indirect control, **6**-39, **6**-40
  linear operations, **6**-35, **6**-36
  nonlinear operations, **6**-36, **6**-37
  operating controls, **4**-4 to **4**-9
  remote control, **4**-11
  slave operation, **4**-11
  small, **4**-38
  digital computers, large-scale, **20**-20
  program (*see* Program control)
  timing and synchronization, **16**-28 to **16**-33
Control unit, digital differential analyzer, **19**-25 to **19**-27
  repetitive analog computer, **6**-24
Control valve, simulation of, **5**-38, **5**-74
Controllers, industrial, **6**-40
  simulation of, **5**-72 to **5**-75
Convair-Astronautics, **4**-40, **4**-44
Convair–San Diego, **4**-40, **4**-57
Conversion errors, **6**-46
Converters, analog-to-digital (*see* Analog-to-digital converters)
  digital-to-analog (*see* Digital-to-analog converters)
  from magnetic tape, **18**-15 to **18**-18
  to magnetic tape, **18**-11, **18**-15
Coordinate conversion unit, rectangular-to-polar, **7**-26 to **7**-28
Coordinate systems for flight equations, **5**-50, **5**-51
Coordinate transformations, with analog computers, **5**-52 to **5**-57, **6**-29
  (*See also* Resolvers)
  in network analogy, **9**-8 to **9**-11
Copper-mandrel potentiometers, **2**-64 to **2**-67
  (*See also* Potentiometers)
Cores (*see* Magnetic cores)
Correlation functions, **5**-94, **5**-95
  measurement of, **5**-96 to **5**-98
Correlators, analog, **5**-96 to **5**-98
  digital, **19**-10, **19**-11
Cosine circuit, **3**-33
Cosine/sine function generator, **7**-23 to **7**-25
Costs of computer operation, **21**-40 to **21**-42
Coulomb, **10**-4
Coulomb friction, simulation of, **5**-40, **5**-45
Counters, binary, **15**-7, **21**-89 to **21**-91
  decimal, **15**-6, **15**-7, **21**-91
  Hollerith, **20**-3
Coupling circuits, digital computer, **10**-7, **11**-34
Creep, analogy for, **9**-86, **9**-87
Cross-correlation function, **5**-94, **5**-95
  analog-computer methods for finding, **5**-98 to **5**-101
  measurement of (*see* Correlators)
Cross-spectral density, **5**-95
  measurement of, **5**-102 to **5**-107

Crossed-fields electron-beam multiplier, **3-40**, **3-41**
Cryotrons, **13-13**
Crystallographic calculations, **21-11**
CSI (*see* Computer Systems, Inc.)
Cubic analog interpolation, **21-70**, **21-71**
Cup and cap, **14-14**
Current-generator circuits, **9-46** to **9-48**, **9-54**
Current stabilizer, **12-86**, **12-87**
Current switch, **3-54**, **3-56**
Cut-product approximation of time delay, **6-5**
Cyclic code, binary, **18-40**, **18-41**
  for scalar product, **17-8**
Cylindrical shell, **9-29**, **9-30**

Data arrangement, **16-6**, **16-7**
Data processing, **21-13** to **21-32**
Data reduction, **21-11**
Data storage, **21-77** to **21-79**
  (*See also* Storage)
D-C amplifiers, design, **2-40** to **2-57**
  examples, **2-72** to **2-79**
  for repetitive analog computers, **6-20** to **6-22**
  for small analog computers, **4-35**
  specifications, **2-40**
  testing of, **4-44** to **4-46**
  transistor, **7-2** to **7-6**, **7-29** to **7-36**
D-C leakage, **2-60**
D-C level restoration, **11-4** to **11-5**, **11-32**
D-C network analyzer, **9-61**
D-C resolvers (*see* Resolvers)
D-C servo, **3-11**
D-C tachometer, specifications, **3-31**
Dead space, simulation of, **3-64**, **3-65**, **3-69**, **5-44**
Decade capacitor, **9-45**
Decimal arithmetic, **15-21**, **20-7**
Decimal code, **14-2**, **14-3**, **15-4**, **15-5**
Decimal counters, **15-6**, **15-7**, **21-91**
Decision integrator, **19-35** to **19-37**
Declarations, **17-27**, **17-35**, **17-36**
Decoder (*see* Digital-to-analog converters)
Decoding nets (*see* Nets)
Decoupled flight equations, **5-49**, **5-58**, **5-59**
Deflection functions, **18-28**
Delay in correlation measurements, **5-96** to **5-98**
Delay circuit, **11-9**, **11-10**, **16-12**
Delay error (*see* Mean-square error)
Delay-line storage, **16-22** to **16-24**
Delay lines, electrical, **10-17**, **10-18**
  mercury and quartz, **12-16**, **12-17**
  notation and symbols, **15-6**
Demodulator for induction resolver, **3-21** to **3-23**
Demodulator noise generator, **3-81**, **3-83**
DeMorgan's theorem, **14-16**
Desampling, **6-46** to **6-48**
Desk-top analog computer, **4-38**, **4-39**
Determinants, evaluation of, **5-138**
DIAD telephone system, **19-13**, **19-14**
Dials, **2-5**
Dian Labs, New York, **4-43**
Dicap, **12-116**
  (*See also* Condenser memory)
Dielectric absorption, transient effects, **2-63**
Dielectric losses, **2-60** to **2-63**
  errors due to, **2-63**
Difference equations, analog solution of, **6-29**, **6-33**, **6-52** to **6-54**
Differential amplifier, **2-46**
Differential analyzer, defined, **1-4**
  digital (*see* Digital differential analyzer)
  electromechanical, **1-4**
  electronic, **1-4**
    representation of transfer functions, **2-28** to **2-40**
    solution of difference equations, **6-52** to **6-54**
  mechanical, **8-9** to **8-13**
Differentiating circuits, **10-7**
Differentiation, electrical, **2-21**
  electromechanical, **8-9**, **8-16**
  by heat transfer, **8-20**
  mechanical, **8-5**, **8-9**
Diffusion equation, analog solution (*see* Heat exchanger; Heat flow)
Digit-driver circuit, **12-87**, **12-88**
Digit-plane winding, **12-55**, **12-56**
Digitac system, **21-81** to **21-84**
Digital-to-analog conversion, machine-tool controls, **21-63** to **21-65**
  coded-digitizer systems, **21-67**, **21-68**
  conversion to analog commands, **21-68**, **21-69**
  quantizers, **21-66**, **21-67**
  stepping devices, **21-65**
Digital-to-analog converters, **6-44**, **7-16**, **18-42** to **18-50**
  number-to-frequency, **18-50**
  number-to-position, **18-46** to **18-48**
  number-to-time, **18-49**
  number-to-voltage, **18-43** to **18-46**
Digital computers, analog computers compared with, **1-4**, **1-5**
  applications, control, **21-62** to **21-94**
    data processing, **21-13** to **21-32**
    management problems, **21-33** to **21-42**
    in science and engineering, **21-2** to **21-13**
    (*See also* Equation solving)
    simulation, **21-42** to **21-62**
  combined with analog computers, **6-42** to **6-52**
  components, **10-3** to **10-56**
  general-purpose, **20-1** to **20-34**
    examples described, **20-10** to **20-19**
    large-scale, **20-19**, **20-20**
    relay types, **20-2** to **20-5**
  special-purpose, **19-1** to **19-74**
    equation solving, **19-2** to **19-5**
    logic computers, **19-5** to **19-9**
    (*See also* Digital differential analyzer)

Digital computers, stored-program (*see* Stored-program computers)
  system design, **16-2** to **16-33**, **21-36**, **21-37**
Digital correlator, **19-10**, **19-11**
Digital differential analyzer, **19-14** to **19-72**, **20-19**
  adder, **19-39**, **19-40**
  address unit, **19-25** to **19-27**
  arithmetic unit, **19-26**
  binary-transfer, **19-27** to **19-29**
  control unit, **19-25** to **19-27**
  decision integrator, **19-35** to **19-37**
  function input, **19-55** to **19-58**
  general-purpose techniques and, **19-67** to **19-72**, **20-19**
  initial conditions, **19-40** to **19-42**, **19-53** to **19-55**
  input and output, **19-53** to **19-59**
  integrator, **19-15**, **19-16**
  integrator interconnection, **19-16** to **19-18**
  multiplication, **19-29** to **19-35**
  number systems, **19-59** to **19-61**
  numerical examples, **19-50** to **19-53**
  output distribution system, **19-22** to **19-27**
  parallel, **19-26**
  principles of operation, **19-15** to **19-29**
  printed output, **19-58**, **19-59**
  programming, **19-61** to **19-67**
  round-off error, **19-46** to **19-48**
  scaling, **19-31**, **19-32**, **19-65** to **19-67**
  serial, **19-18**, **19-19**
  servos, **19-35** to **19-39**
  signum function, **19-39**
  total error, **19-48** to **19-50**
  truncation error, **19-42** to **19-46**
  variable multiplication, **19-32** to **19-35**
Digital EDC (*see* Economic-dispatch computers)
Digital interpolation, **21-72** to **21-77**
Digital-operational systems, **21-89** to **21-94**
Digital servo, **3-58**, **7-15** to **7-18**
Digital simulation (*see* Simulation)
Digital sorting (*see* Sorting)
Digits, binary, **14-2**, **15-2**
  symbols, **17-2**
DINA digital analyzer, **19-4**, **19-5**
Diode(s), **10-10**, **10-11**, **10-22** to **10-29**
  alloyed-junction, **10-28**
  for analog computers, specifications, **3-63**
  breakdown voltage, **10-24**
  characteristics (ferromagnetic shift registers), **12-15**, **12-16**
  crystal, **10-22** to **10-29**
  diffused-junction, **10-29**
  gates, **13-1**, **13-4**, **13-5**
  impedance, **10-26**
  point contact, **10-27**
  silicon, specification (condenser memory), **12-120**
  switching times, **10-26**
Diode(s), testing, **4-53**, **4-54**, **10-25**
  transient behavior, **10-25**
  vacuum, **10-10**, **13-1**
Diode-bridge switch, **6-24**, **6-26**, **6-28**
Diode-capacitor memory, **12-116** to **12-126**
Diode choppers, **7-6**
Diode coincidence circuits, **13-1**, **13-2**, **13-4**
Diode function generators, **3-66** to **3-68**, **3-72** to **3-74**, **3-79**, **5-18**, **6-18**, **6-19**
  automatic setting, **3-68**, **4-13**
  block diagram symbol, **1-8**
  control-system applications, **6-36** to **6-38**
Diode limiters (*see* Limiter circuits)
Diode multiplier (*see* Quarter-square multiplier)
Diode-ring modulator, **7-6**, **7-8**, **7-9**, **7-13**, **7-14**
Diode switches, **12-79**, **13-9**, **14-20**
Direct-analogy computers, **9-3** to **9-87**
  active elements, **9-45** to **9-49**
  conductive-liquid analogies, **9-82** to **9-84**
  conductive-solid analogies, **9-80** to **9-82**
  defined, **9-3**
  errors, **9-30** to **9-35**
  excitation, **9-51** to **9-54**
  general requirements, **9-36**
  hydrodynamic and hydraulic analogies, **9-86**, **9-87**
  installations, size of, **9-60**
  location of components, **9-58**
  loop and nodal analogies, **9-3** to **9-6**, **9-46**
  membrane analogies, **9-84**, **9-85**
  metering, **9-54** to **9-58**
  nonlinear elements, **9-49** to **9-51**
  operation, **9-59**, **9-60**
  output devices, **9-57**
  passive elements, **9-37** to **9-45**
  passive-network analogies, realizability of, **9-7**
  resistance-network analogy, **9-69** to **9-80**
  structural analogies, dynamic, **9-12** to **9-19**
  static, **9-19** to **9-30**
Direction cosines, **5-59**, **5-60**
Disjunctive normal forms, **14-15**
Disk, magnetic, **12-18**, **12-33**, **12-34**
Displays (*see* Symbol displays)
Dissipation factor of capacitor, **2-60** to **2-63**
Distillation column, simulation of, **5-80** to **5-82**, **6-33**
Distortion, **11-15**
Dither (*see* Smoothing oscillator)
Division, digital, decimal, **15-21**
  serial binary, **15-15**, **15-16**
  electronic analog, **3-56**, **3-58**
    with servomultipliers, four-quadrant, **5-7**, **5-11**
    two-quadrant, **5-8** to **5-11**
    with transistor multipliers, **7-18** to **7-20**
Donner Scientific Company, **3-55**, **3-63**
  Model 35 diode function generator, **3-74**
  Model 3101, **2-73**, **2-77**
Double integral, DYSTAC computation of, **6-33**
Double registers, **10-21**

Douglas Aircraft Co., Quadratron, **3-75** to **3-77**
Drift, **6-39**
  compensation in d-c amplifiers, **2-51**
  in transistor d-c amplifiers, **7-2** to **7-6**, **7-29** to **7-36**
Drivers, **11-7**
  digit-plane, **12-87**, **12-88**
  gate, **13-5**, **13-6**
  for magnetic-core circuits, **12-16**
  relay, **11-7**
  transformer-coupled, **11-36**, **11-37**
Drum, digital differential analyzer, **19-20**, **19-21**
  plotter, **18-54**
  symbol display, **18-21**
  (*See also* Magnetic-drum computers)
Drum correlator, **5-97**
Dual-control-grid tubes, **13-3**, **13-4**
Duality, **14-16**
  of analogies, **9-5**
Dynamic errors, in amplifiers, **2-57**, **2-58**
  in servomechanisms, **3-18**, **5-37**, **5-40**
Dynamometer multiplier, **3-40**
DYSTAC (Dynamic Storage Analog Computer), **6-32** to **6-34**

Earth Coordinate System (ECS), **5-50**
EASE (*see* Beckman/Berkeley)
Eccles-Jordan transistor flip-flop, **12-4**
Economic-dispatch computers (EDC), **5-155** to **5-159**
  digital, **5-159**
EDVAC, **21-2**
Eigenvalue problems, analog solution, **5-112**, **5-113**, **5-116**
  digital solution, **21-6**, **21-7**
Elastic stop (*see* Limit stops)
Electric networks, digital simulation, **21-60**, **21-61**
Electric power for computers (*see* Power supply)
Electric-power generation, digital control, **21-86** to **21-89**
Electrical terms, definition of, **10-4**, **10-5**
Electrolytic tanks, **9-81** to **9-84**
Electromechanical (a-c) analog computers, **8-16** to **8-18**
Electron-beam multiplier, **3-40**, **3-41**
Electron-trajectory model, **9-84**, **9-85**
Electronic Associates, Inc., **2-80**, **3-23**, **3-56**, **4-43**
  Model 6.002, **2-73**, **2-78**
  Simulag (*see* Simulag)
Electronic differential analyzer (*see* Differential analyzer)
Electronic Industries Association, tape dimensions, **18-2**, **18-3**
Electronic musical instrument, digital, **19-12**
Electronic resolvers (*see* Resolvers)
Elgenco, Inc., **3-83**
Emitter followers, **11-2** to **11-4**
Encoder (*see* Analog-to-digital converters)
"End-around carry," **15-11**
End-pulse circuit, **11-13**
End resistance, **2-3**, **2-4**, **3-4**, **3-5**
Energy-release motors, **18-47**
Engineering applications of digital computers, **21-2** to **21-13**
ENIAC, **12-2**, **21-2**
Ensemble averages, analog computation of, **5-94**, **6-31**
Equation solvers, analog, **5-133** to **5-149**
  defined, **1-4**
Equation solving, analog, algebraic equations, **5-133** to **5-139**
  beam equations, **5-116** to **5-119**, **5-129** to **5-132**
  direct-analogy computers, **9-12** to **9-19**, **9-37**
  linear differential equations, **5-13** to **5-15**
  nonlinear differential equations, **5-15** **5-16**
  partial differential equations, **5-110** to **5-132**
  polynomial roots, **5-139** to **5-141**
  resistance-network analogy, **9-69** to **9-80**
  digital, **6-43**
  differential equations, numerical solution, **21-50** to **21-52**
  real-time solution, **21-52** to **21-54**
  equation-solving controls, **21-81**
  integro-differential equations, **21-7**, **21-8**
  linear equations, **19-2**, **19-3**, **21-2** to **21-4**
  nonlinear algebraic equations, **21-8**
  ordinary differential equations, **21-4** to **21-6**
  partial differential equations, **19-4**, **19-5**, **21-8** to **21-10**
  quadratic equation, **17-15**, **17-16**
  simultaneous linear algebraic equations, **19-2** to **19-4**
  electronic differential analyzer, **6-52** to **6-54**
Equivalent circuit, of amplifier stage, **2-45**, **2-46**, **2-47**
  of capacitor, **2-60** to **2-63**
  of potentiometer, **2-4**, **2-64**
  of resistor, **2-59**
Ergodic processes, **5-95**
Error-detecting codes, **14-4**, **14-6**, **14-7**
Errors, in amplifiers, **2-57**, **2-58**
  in analog-to-digital conversion, **6-46**
  digital differential analyzer, **19-42** to **19-53**
  round-off, **19-46** to **19-48**
  total, **19-48** to **19-50**
  truncation, **19-42** to **19-46**
  direct-analogy computers, **9-30** to **9-35**
  mean-square (*see* Mean-square error)
  in mechanical analog computers, **8-2**, **8-3**
  in network analog computers, **9-30** to **9-35**, **9-59**, **9-60**
  in servomechanisms, **3-18**, **5-37**, **5-40**

Euler angles, **5-67** to **5-69**
  (*See also* Flight equations)
Excess-three binary code, **14-3**, **15-4**, **15-5**
Excess-50 representation, **17-4**
Excitation system for network analog computer, **9-51** to **9-54**
EXCLUSIVE OR circuit, **13-6**
Expressions, arithmetic, **17-4**, **17-5**, **17-31**, **17-32**
  Boolean, **17-4**, **17-5**, **17-32**, **17-33**
  Boolean vector, **17-5**
  conditional, **17-33**, **17-34**
  designational, **17-34**
  switches, **17-34**
External compensating networks, **2-51**

Fabrication error, **8-2**
Fairchild Engine and Airplane Co., **19-2**, **19-3**
Farad, **10-4**
Fault studies, **9-63**, **9-68**
Feedback, negative or inverse, **11-17**, **11-18**
Feedback control system, basic, block diagram, **6-35**
  defined, **5-35**
Feedback limiter, **3-62**, **3-65**
  block-diagram symbol, **1-10**
Feedback regulation of multiplier gain, **3-51**
Ferranti truth-table computer, **19-6**
Ferrite cores, **10-21**, **10-22**
  materials, properties, **12-44**, **12-45**
  typical, **12-47** to **12-51**
Ferroelectric switches, **12-114**, **12-115**, **13-11** to **13-13**
Ferroelectrics, **12-106** to **12-116**
Ferrography, **18-53**
Ferromagnetic shift register, **12-5** to **12-16**
Ferrospinels, **12-44**
Fields of data, **21-19** to **21-21**
Files, tape, **21-19**, **21-20**
Film potentiometer, **3-8**, **3-9**
  for resolver, **3-23**, **3-25**
Film resolver, **3-23**, **3-25**
Finite-difference approximations, **9-30** to **9-34**
Fixed-point arithmetic, **15-8** to **15-21**, **20-6**, **20-7**
Flexowriter, **18-3**
Flight equations, **5-49** to **5-60**
Flight simulators, **5-64**, **21-56** to **21-59**
Flight tables, **5-65** to **5-71**
Flight trainers, **5-65**, **21-56** to **21-59**
Flip-flop, **12-3**, **12-4**, **16-12**
  notation, **15-5**
Flip-flop shift register, **16-14**
Floating-point arithmetic, **15-22** to **15-24**
  in computer classification, **20-6**, **20-7**
  using analog computing elements, **6-29** to **6-31**
Floating-point numbers, **15-22** to **15-24**, **17-4**, **17-31**
Flood control, **6-41**
Flow, simulation of, **5-70**, **5-78**
Flow diagrams, programming, **17-8** to **17-10**
Flow transmitter, simulation of, **5-73**
Fluid-flow problem, **19-53**, **19-54**
Fluid mapper, **9-86**
FM-AM multiplier, **3-48**, **3-52**
Follow-up potentiometer, **3-3**
Force-current analogy, **9-3** to **9-6**
Force-voltage analogy, **9-3** to **9-6**
Four-address computer, **16-10**, **16-11**, **20-27**
  command system, **17-24**
Four-diode switch, **6-26**, **6-28**
Fourier analysis, analog methods, **6-54** to **6-56**
Fourier coefficients, **5-114**, **5-115**
Fourier inversion, **5-105** to **5-107**
Franklin Institute, **6-29**
Frequency converter, **8-21**
Frequency dependence of capacitance, **2-62**, **2-63**
Frequency modulation, tape recording, **6-11**, **6-12**
Frequency-to-number converters, **18-41**, **18-42**
Frequency response of electronic multipliers, **3-54**
Frequency stabilization, **2-49**
Friction, simulation of, **5-40**, **5-45**
Fuel-cost setter, **5-155**
Function approximation, digital, **21-44** to **21-49**
Function cams, **8-2**, **8-6**, **8-7**
Function generators, analog, **3-66** to **3-80**, **5-18**, **5-19**
  cathode-ray-tube, **3-68** to **3-70**, **3-73**, **3-74**, **3-79**, **3-80**
  mechanical, **8-5** to **8-9**, **8-11** to **8-13**
  in network analog computers, **9-49** to **9-51**
  transistor, **7-20** to **7-29**
  varistor, **3-73**, **3-75** to **3-77**
  (*See also* Diode function generators)
  digital, **6-52**
  hybrid analog-digital, **3-75**, **3-80**, **6-30**, **6-52**
Function input, digital differential analyzer, **19-55** to **19-58**
Functions, programming, **17-7**, **17-28**, **17-29**, **19-29**
  of two or more variables, **3-78** to **3-80**

Gain, amplifier, measurement of, **4-44**, **4-45**
Galvanometer, **18-22**, **18-53**, **18-54**
Game theory, **21-35**
Gas-flow process with analog control, **6-36**, **6-38**, **6-39**
Gates, **13-1** to **13-6**
  coincidence, **13-1** to **13-6**
  with memory, **13-6**, **13-7**
  crystal-diode, **13-4**, **13-5**
  gate-driver combination, **13-5**, **13-6**
  vacuum-tube, **13-1** to **13-3**
  (*See also* AND gate; OR gate)
Gating amplifiers, **12-125**
Gating pulses, time of, **16-26**

Gauss effect, **3**-41, **7**-18
Gauss-Seidel iteration, **5**-134, **5**-135, **19**-2, **19**-3
Gaussian noise, **5**-96, **5**-99, **5**-109
Gaussian-noise generators, **5**-96
Gaussian random process, **5**-94
Gears in computer servos, **3**-13, **3**-31
Geiger-Mueller counter, noise generator, **3**-81, **3**-82
General Electric Company, EDC, **5**-158, **5**-159
  network analyzer, **9**-61, **9**-64 to **9**-66
General-purpose computers (*see* Digital computers, general-purpose)
Generator for scalar-product linear code, **17**-19
Generator-load-setter unit in network analyzer, **9**-63
Geometrical models, coding, **14**-5, **14**-6
Gimbal, **5**-66
Gimbal computation, **5**-67, **5**-68
Gimbal lock, **5**-66
Goodyear Aircraft Corporation, **3**-11, **3**-29, **3**-82, **4**-43
  Model GEDA A-14, **2**-73
GPS Instrument Company, Inc., special repetitive-computer components, **6**-19
Gradient method (*see* Steepest-descent methods)
Grand Central Station computer classification system, **20**-5, **20**-6
Granularity, simulation of, **5**-40, **5**-45
Grid-blocking capacitor in d-c amplifier, **2**-56, **2**-79, **6**-21
Grid current in d-c amplifier, **2**-56
Ground circuits for analog computers, **4**-31, **4**-32
Group alphabets, **14**-7, **14**-8, **14**-10, **14**-11
Guided missiles (*see* Missiles)
Gyro-controlled flight tables, **5**-67, **5**-68

Half-adder, **14**-18
  serial, **15**-8, **15**-9
Hall effect, **3**-41, **7**-18, **7**–19, **9**-63
Hall-effect multiplier, **3**-41, **7**-18, **7**-19
Haller, Raymond & Brown computer, **19**-2, **19**-3
Hamming alphabets, **14**-8 to **14**-10
Harmonic analysis of periodic functions, **6**-56
Heat balance, **5**-84
Heat exchanger, simulation of, **5**-79 to **5**-82, **5**-91 to **5**-93, **6**-29, **6**-30
Heat flow, analog representation, difference methods, **5**-119 to **5**-122, **5**-126 to **5**-128
  network analogies, **9**-77, **9**-78
  in nuclear reactor, **5**-91, **5**-92
  into reactant, **5**-77
  separation of variables, **5**-110 to **5**-112
  (*See also* Heat exchanger)
Heat transfer, simulation of, **5**-91 to **5**-93
Heat-transfer computing elements, **8**-18 to **8**-22
Heat-transfer transducers, **8**-18 to **8**-20
  self-heating, **8**-20 to **8**-22
Helix-bar recorder, strobed, **18**-56, **18**-57
Henry, **10**-5
Heterodyne-type spectrum analyzer, **5**-103 to **5**-105
Hewlett-Packard function generator, **3**-75, **3**-77
Hexadecimal number system, **15**-2 to **15**-4
High-frequency stabilization, **2**-49
High-gain amplifier, block-diagram symbol, **1**-7
HOLD mode, of analog integrator, **4**-4 to **4**-8, **6**-32 to **6**-34
  of continuous resolver, **3**-30
Hollerith counter, **20**-3
Homing missile, **5**-61, **5**-62
Human behavior, digital simulation of, **21**-61, **21**-62
Hybrid analog-digital computers, **1**-5, **6**-26, **6**-29 to **6**-34, **6**-42 to **6**-52
  (*See also* Digital differential analyzer)
Hybrid analog-digital function generator, **3**-75, **3**-80, **6**-30, **6**-52
Hydraulic analogy, **9**-86, **9**-87
Hydraulic servo, simulation of, **5**-38
Hydrodynamic analogy, **9**-85 to **9**-87

IBM decision machine, **19**-6
IBM 370 printer, **18**-7
IBM 407 accounting machine printer, **18**-6
IBM 650, **20**-10 to **20**-12
IBM 701–704, card-sensing circuits, **18**-11
IBM 720A printer, **18**-7
IBM 727, **6**-14
IBM Type 1402 Card Read-Punch, **18**-9, **18**-10, **18**-12
IBM 1403 printer, **18**-7
Ideal response, amplifier stage, **11**-1, **11**-2
Identifier, **17**-4, **17**-26
Illinois, University of, coincidence gate with memory, **13**-6, **13**-7
Image-converter tubes, **18**-28, **18**-29
Implicit computation, analog, **5**-7 to **5**-12
Incremental converters, **18**-46, **18**-47
Incremental fuel cost, **5**-156
Incremental production cost, **1**-155
Incremental techniques, **18**-35 to **18**-37
Independent linearity, **3**-5
Index register, **17**-11, **17**-12, **20**-27 to **20**-29
  single-address system with, **16**-11, **20**-28
Indexing, **17**-7, **17**-27
Indicators, strobed, **18**-22, **18**-23
Inductance, **10**-5
Induction resolver, **3**-19 to **3**-23, **8**-16, **8**-17
  (*See also* Resolvers)
Inductors for network analog computers, **9**-39, **9**-41 to **9**-44, **9**-62, **9**-63
Industrial controllers, simulation of, **5**-72 to **5**-75, **6**-40
Inelastic stop (*see* Limit stops)
Information, parallel and serial representation, **16**-13, **16**-14
  (*See also under* Parallel; Serial)

Information, periodicity of flow, **16**-14, **16**-15
  transfer of, and access time, **16**-15, **16**-16
    between memories, **16**-16, **16**-17
  translation from serial to parallel, **16**-14
Information coding (*see* Coding)
Inhibit winding, **12**-55, **12**-56
Initial conditions, scaling, **5**-7
Initial-value problems, **21**-4 to **21**-6
Input circuits for computer servos, **3**-14, **3**-15
Input impedance, of amplifier stage, **2**-48
  of operational amplifier, **2**-17
Input and output, analog, for random processes, **5**-94, **5**-95, **5**-99 to **5**-102
  digital, **18**-1 to **18**-58
    ALGOL 60, **17**-29
    command list, **17**-13, **17**-14
    control, **16**-5 to **16**-7
    relay types, **20**-5
  digital differential analyzer, function input, **19**-55 to **19**-58
    initial-condition input, **19**-53 to **19**-55
    output distribution system, **19**-22 to **19**-27
    printed output, **19**-58, **19**-59
Input-output programmer, analog, **4**-13, **4**-15
Instructions (*see* Commands)
Integers, **17**-4
Integrating servo, **3**-29
Integrator, analog, defined, **1**-4
  electronic, **5**-7
    block-diagram symbol, **1**-7
    heat-transfer, **8**-20
    reset circuits, **6**-22 to **6**-25, **6**-27, **6**-28
  mechanical, **8**-2, **8**-4, **8**-5, **8**-10
  modes, **4**-4 to **4**-8
  digital differential analyzer, **19**-15, **19**-16
  decision, **19**-35 to **19**-37
  interconnections, **19**-16, **19**-17, **19**-42 to **19**-50
Interconnections, analog computers, **4**-30 to **4**-33
Interlacing, Royal-McBee LGP-30, **20**-15
Interlocking circuits, **16**-5
Interpolation, analog, **3**-78, **3**-79
  cubic, **21**-70, **21**-71
  linear, **21**-69, **21**-70
  digital, **21**-72 to **21**-77
  in machine-tool control, **21**-69 to **21**-77
  spline, **21**-71, **21**-72
Interpolator, cubic, **21**-70, **21**-71
  spline, **21**-71, **21**-72
Interpretive system, **17**-21, **17**-22
Interrupt logic, **20**-30 to **20**-33
Interstage network, d-c amplifier, **2**-48, **2**-49
Inventory control, **21**-15 to **21**-17
Inverter, buffer (*see* Buffer inverters)
  circuit, **16**-12
  notation, **15**-5, **15**-6
Inverting gates, Bendix G-15, **20**-15, **20**-16, **20**-18
IRE standard symbols for analog computers, **1**-7 to **1**-10
Isograph, **5**-140, **5**-143, **5**-144
I.T.E. Circuit Breaker Co., **9**-61
Item (information unit), **17**-4
Item (unit record), **21**-19

Jet-engine test facility, computer control of, **6**-38, **6**-39
Jevons logic computer, **19**-5
Joule, **10**-4
Jump (*see* Transfer of control)
Junction, programming, **17**-8
Junction transistor (*see* Transistor)

Kalin-Burkhart logical truth calculator, **19**-5, **19**-6
Karnaugh magnetic switch, **12**-8, **12**-9
Keys, data, **21**-19 to **21**-21
Kidde Warehouse System, **21**-40
Kirchhoff's laws, **10**-5, **10**-6

Lag-lead networks, **2**-49 to **2**-51
Language (*see* ALGOL 60; Alphabets; Coding; Commands; Symbols)
Laplace's equation, analog solution, **9**-69 to **9**-76, **9**-80 to **9**-86
Large-scale digital computer, **20**-19 to **20**-33
Leakage, in capacitors, **2**-60
  in patch cords, **4**-29
  in patchboards, **4**-27 to **4**-30
Least-square polynomial approximation, **21**-45 to **21**-47
LGP-30, **20**-12 to **20**-16
Limit stops, for computer servos, **3**-14, **3**-27
  simulation of, **5**-43, **5**-46 to **5**-49
Limiter circuits, basic diode, **3**-62 to **3**-66
  bridge limiter, **3**-66
  comparators, multivibrators, and others, **3**-63, **3**-70, **3**-71
  precision, **3**-62, **3**-63, **3**-67, **3**-69
  symbols, **1**-10
  transistor function generator, **7**-20, **7**-21
  transistor multiplier, **7**-10, **7**-14, **7**-15
Line printer, **18**-6 to **18**-8
Line unit, **9**-62
Linear analog interpolation, **21**-69, **21**-70
Linear code, scalar product with, **17**-19, **17**-20
Linear control operations, analog, **6**-35, **6**-36
Linear equations (*see* Equation solving)
Linear method of reading, **12**-58 to **12**-60
Linear-programming problems, analog-computer solution, **5**-149 to **5**-152
Linearity of potentiometers, **2**-3, **2**-4, **3**-4 to **3**-8
Linearization of control characteristics, **6**-37, **6**-38
Linkages, **8**-2, **8**-3, **8**-8, **8**-9

Load-equalizing resistors in servomultipliers, 1-9, 3-8, 5-8, 5-10, 5-12
Load-flow studies, 5-155 to 5-159, 9-68
Load resistance, effect on d-c amplifier, 2-46, 2-47
Load-setter unit, 9-63
Load unit, 9-62
Loading of potentiometers (*see* Potentiometer loading)
Loading slide rule, 5-155, 5-156
Logarithmic multiplier, 3-41, 3-42
Logic, in circuit design, 14-17, 14-18
  interrupt, 20-30 to 20-33
Logical circuits, elementary, 16-11, 16-12
Logical computer, 19-5 to 19-9
Logical section, IBM 650, 20-10, 20-11
Loop, magnetic-drum computer, 20-8, 20-9 (*See also* Register, circulating)
Loop analogy, 9-3 to 9-6, 9-46
Loops, minor, in magnetic storage cores, 12-56 to 12-58, 12-67, 12-68
  transfer, in shift registers, 12-7 to 12-10
Lukasiewicz notation, 19-7, 19-8

Machine-tool control, 21-63 to 21-81
Magnetic amplifiers, 3-11
Magnetic cards, 18-19
Magnetic cores, 10-18 to 10-22, 12-14, 12-15, 12-41 to 12-45
  adverse environments, 12-100, 12-101
  bidirectional switches, 12-77, 12-78
  binary switches, 12-68
  characteristic curves, 12-15
  characteristics, 12-14, 12-42
  coincident-current selection, 12-52 to 12-55
  coincident-current storage, 12-80, 12-81
  control by switch, 12-56
  drivers for, 12-16
  ferrites, 10-21, 10-22, 12-44, 12-45
  geometry, 12-45 to 12-47
  inhibit winding, 12-55
  logical representation, 12-6
  maintenance, 12-102, 12-103
  materials, 12-42 to 12-45
  matrices, 12-55
  metal tapes, 12-43, 12-44
  minor loops, 12-56 to 12-58, 12-67, 12-68
  nondestructive read-out, 12-103 to 12-105
  read amplifier, 12-88 to 12-90
  recovery times, 12-16
  saturated loops, 12-57, 12-58
  sense windings, 12-60 to 12-64
  stability, 10-22
  storage, 12-41 to 12-106
  switch drive, 12-83, 12-84
  switches, 12-8, 12-9, 12-64 to 12-80
  switching concepts, 12-5, 12-6
  switching times, 12-42, 12-43
  system, 12-80 to 12-106
  technical data, 10-20, 10-21
  testing, 12-51, 12-52, 12-101, 12-102
  thin films, 12-44
Magnetic cores, transfer loops, 12-7 to 12-10
  transformer drive, 12-82, 12-83
  types, 12-43 to 12-45
  unidirectional switches, 12-71 to 12-77
  word selection, 12-58 to 12-60
Magnetic disk, 12-18, 12-33, 12-34
Magnetic-drum computers, analog, 6-14 to 6-16
  digital, 6-14 to 6-16, 12-18 to 12-20, 12-31, 12-32, 20-10 to 20-19
    with digital differential analyzer, 19-67 to 19-71
    examples described, 20-10 to 20-19
    special characteristics, 20-7 to 20-10
  hybrid analog-digital function generator, 6-30
Magnetic head, 12-21 to 12-31
Magnetic memory systems, 12-17 to 12-34, 18-14, 18-15
Magnetic page writer, 18-19
Magnetic recording, analog, 6-13 to 6-16, 18-13, 18-14, 21-77 to 21-79
  machine-tool control, 21-77 to 21-79
  digital, 12-18 to 12-20, 12-28 to 12-31, 18-14 to 18-18
    of analog data, 6-13
    converters, from magnetic tape, 18-15 to 18-18
    to magnetic tape, 18-11, 18-15
    input and output devices, 18-15, 18-18 to 18-20
    non-return-to-zero (NRZ), 12-29, 12-30
    phase-modulation, 12-30, 12-31
    pulse, 12-30
  rotating-head system, 18-18, 18-19
Magnetic shift register (*see* Shift register)
Magnetic switching (*see* Switches)
Magnetic tape (*see* Magnetic recording; Tape)
Magnetoresistance, 7-18
Magnetoresistor multiplier, 3-41, 7-18
Maintenance of analog computers, 4-44 to 4-58
  records, 4-56 to 4-58
Management problems, 21-33 to 21-42
  control, 21-33, 21-37
  strategy, 21-33, 21-36
  systems design, 21-36, 21-37
  systems operation, 21-37 to 21-40
  tactics, 21-33, 21-36, 21-37
Management Sciences Research Project, University of California, Los Angeles, 21-36, 21-37
Mapping, analog method, 6-56
  digital differential analyzer, 19-61 to 19-65
MARK TRANSFER, 17-22
Massachusetts Institute of Technology, hybrid analog-digital computer, 6-29 to 6-31
  network analyzer, 9-61
  numerically controlled milling machine, 21-72 to 21-75
  repetitive analog computer, 6-22

Material balance, 5-84
Mathieu's equation, 19-53, 19-54
Matrix (matrices), buffer storage unit, 12-93 to 12-97
  coincident-current, 12-53, 12-54
  in parallel, 12-55, 12-56
  and sense windings, 12-60 to 12-64
  triple-coincidence, 12-54, 12-55
Matrix computations, digital, 21-2 to 21-4
Matrix inversion, 5-138, 5-139
Maximum-effort servo, 5-39
Maximum-remembering circuit, 3-72
Mean-square error, 5-40, 5-94
  analog computation of, 5-95 to 5-101
Mechanical analog computers, 8-2 to 8-16
Mechanical differential analyzer, 8-9 to 8-13 (*See also* Integrator, analog, mechanical)
Membrane analogy, 9-84, 9-85
Memory, digital computer, 12-1 to 12-126
  access time, 16-15, 16-16
  addressing, 16-5
  Bendix G-15, 20-15
  circuits, 16-12, 16-13
  coincident current, 12-80
  condenser, 12-116 to 12-126
  diode-capacitor system, 12-116 to 12-126
  ferroelectrics, 12-106 to 12-116
  ferromagnetic shift registers, 12-5 to 12-16
  flip-flops, 12-3, 12-4
  IBM 650, 20-10
  large-scale, 20-19
  magnetic-core storage and switching, 12-41 to 12-106
  magnetic-drum, 20-7, 20-8
  magnetic memory systems, 12-17 to 12-34
  page, 20-30 to 20-33
  permanent, 16-14, 16-15
  recirculating, 16-12, 16-13
  Royal-McBee LGP-30, 20-12
  timing and synchronization, 16-14 to 16-16
  transfer of information between, 16-16, 16-17
  transient, 16-14, 16-15
  Williams-tube system, 12-34 to 12-41
  (*See also* Storage)
Memory pair, analog, 6-32 to 6-34, 6-54
Mercury memory systems, 12-16, 12-17
Merge command, 17-25
Merging (sorting), 21-24, 21-29 to 21-33
Metering, in a-c network analyzers, 9-62, 9-64 to 9-66
  in direct-analog computers, 9-54 to 9-58
  in electronic analog computers, 4-9, 4-10, 4-12 to 4-15, 4-38
Mica capacitors, 9-44
Microprogramming, 16-32, 16-33
Microwave circuit, 11-37
Military supply system, 21-40
Minimality, 14-18 to 14-20, 14-22
Minimum cycling, 5-71
Minimum-excursion indicator, 4-20, 4-21
Minimum overshoot, 5-71
Minimum-remembering circuit, 3-72
Minor cycle, 16-30, 16-31
Missiles, guidance, 5-60 to 5-62, 5-64, 5-65
  block diagram, 5-50
  combined analog-digital simulation, 6-49 to 6-52
  training devices, 5-65
  trajectory equations, 6-43
  (*See also* Aircraft)
MIT (*see* Massachusetts Institute of Technology)
Mixing of liquids, simulation of, 5-76
Möbius bands, 18-22
Modes, analog computer operation, 4-4 to 4-9
Modified-analog network analyzer, 9-61
Modulator, diode-ring, 7-6, 7-8, 7-9, 7-13, 7-14
  electromechanical and semiconductor compared, 7-3, 7-4
  for induction resolver, 3-21
  (*See also* Chopper)
Monoscopes, 18-27
Moscow, University of, SETUN, 15-4
Motors, number-to-position converters, 18-46 to 18-48
Multiplexing of analog computing elements, 6-27, 6-28
Multiplication, analog, 1-8, 1-9, 5-8, 5-9, 9-49
  by constant coefficients, 5-6, 5-7
  mechanical, 8-5, 8-10
  with servomultipliers, 5-5, 5-8, 5-9
  digital, binary, 15-2
  decimal, 15-21
  floating-point, 15-24
  hexadecimal, 15-3
  octal, 15-3
  Royal-McBee LGP-30, 20-14
  serial binary, 15-13, 15-14
  stored-carry, 15-18, 15-19
  digital differential analyzer, 19-29 to 19-35
Multipliers, analog, AM-AM, 3-46, 3-47
  AM-FM, 3-48, 3-52
  AM-PM, 3-48 to 3-50, 7-12 to 7-14
  division using, 3-56
  feedback regulation of gain, 3-51 to 3-56
  logarithmic, 3-41, 3-42
  mechanical, 8-5, 8-10
  in network analog computers, 9-49
  PAM-PWM (time-division), 3-50 to 3-57, 7-9 to 7-14
  quarter-square, 3-42, 3-43, 3-76
  testing of, 4-50 to 4-53
  transistor, 7-9 to 7-20
  triangle-integration, 3-42 to 3-46, 7-14, 7-15
  vector resolution and composition using, 3-59, 3-60
  digital, binary operational, 21-74, 21-75, 21-89 to 21-91
  four-channel, 15-20, 15-21

Multipliers, digital, serial binary, **15**-13, **15**-14, **15**-16
stored-carry, **15**-18, **15**-19
whiffletree, **15**-20
Multivibrator, analog, **3**-63, **3**-71
digital, **11**-11 to **11**-13
Musical instrument, digital electronic, **19**-12
Mylar capacitor, **2**-55, **2**-61, **9**-44
Mylar magnetic tape, **12**-32

NACA interpolator, cubic, **21**-70, **21**-71
spline, **21**-71, **21**-72
National Bureau of Standards (*see* SWAC)
Negative impedance, **9**-48
Negative numbers, **15**-11, **15**-14, **15**-15
Net-interchange setter, **5**-155
Nets, **14**-18 to **14**-23
with delays, **14**-22, **14**-23
Network analogies (*see* Direct-analogy computers)
Network analyzers, power-system, **9**-61 to **9**-68
Newton's method, **17**-15 to **17**-17
Node analogy, **9**-3 to **9**-6, **9**-46
Noise, in amplifier, **11**-16, **11**-17
in bistable magnetic cores, **10**-21
in d-c amplifiers, **6**-39
Gaussian, **5**-96, **5**-99, **5**-109
in potentiometers, **3**-6, **3**-8
random, simulation of, **21**-59, **21**-60
in servo amplifiers, **3**-9
white, **5**-96, **5**-99 to **5**-101
Noise generation in digital computers, **12**-11 to **12**-13
Noise generators, **3**-81 to **3**-83, **5**-108, **5**-109, **6**-19, **6**-31
Gaussian, **5**-96
Nonlinear analytic approximations with function generator, **5**-18, **5**-19
Nonlinear differential equations (*see* Equation solving)
Nonlinear spring, **9**-50
Nonlinear transfer characteristics, simulation of, **5**-42, **5**-43
Non-return-to-zero (NRZ) recording, **12**-29, **12**-30
Normal coordinates, **5**-118
Normal modes, **5**-112 to **5**-114
Normalization, **15**-23
Normalize command, **17**-25
NOT, **14**-13, **15**-5, **17**-2, **17**-36
Notation, for analog computers, **1**-6, **5**-5
coding, **17**-2, **17**-3
binary, **15**-5, **15**-6
(*See also* Alphabets; Coding)
Notch filter using operational amplifiers, **2**-40
Nuclear kinetics, simulation of, **5**-87
Nuclear reactors, simulation of, **5**-87 to **5**-93
Nulling-type equation solvers, **5**-139 to **5**-141, **5**-144
Number-to-boxcar-to-voltage converters, **18**-44 to **18**-46
Number-to-frequency converters, **18**-50
Number-to-position-to-voltage converters, **18**-45, **18**-46
Number sieve, **19**-11, **19**-12
Number-to-time converters, **18**-49
Number-to-voltage converters, **18**-43 to **18**-46
Number-to-voltage-to-position converters, **18**-48
Numbers, ALGOL 60, **17**-31, **17**-32
binary (*see* Binary numbers)
Boolean, **17**-4, **17**-5
digital differential analyzer, **19**-59 to **19**-61
floating-point, **15**-22 to **15**-24, **17**-4, **17**-31
hexadecimal, **15**-2 to **15**-4
negative, **15**-11, **15**-14, **15**-15
octal, **15**-2, **15**-3
ternary, **15**-4

Octal number system, **15**-2, **15**-3
Odometer, **18**-21
Offset voltage, d-c amplifier, **2**-51
Ohm, **10**-4
Ohm's law, **10**-5
One-address computer, **16**-7, **16**-8, **20**-20 to **20**-23
command system, **17**-23, **20**-21, **20**-23, **20**-24
with address modification, **20**-28
with index registers, **16**-11, **20**-28
One-shot multivibrator, **11**-11 to **11**-13
Open-loop gain, measurement, **4**-44, **4**-45
Open-shop operation of analog computers, **4**-47
OPERATE mode, **4**-4
Operational amplifiers (*see* Amplifiers, analog computer)
Operational flight trainer (OFT), **5**-65
Operational multiplier, **21**-89 to **21**-94
accuracy, **21**-93, **21**-94
Operational relays (*see* Relays)
Operators, of analog computers, **4**-41, **4**-42
in partial system tests, **5**-64, **5**-65
Operators (operational symbols), **17**-2, **17**-3, **17**-36
Optimalizing control, **6**-42
OR circuit, **16**-12
EXCLUSIVE, **13**-6, **14**-13, **16**-12
negative, **13**-1, **13**-2, **13**-4
positive, **13**-1, **13**-2, **13**-4
OR gate, crystal-diode, **13**-4, **13**-5
notation, **15**-5
parametric amplifier, **13**-14
ORACLE computer, **19**-4
Order code (*see* Commands)
Oscillator, parametric, **11**-37, **11**-38, **13**-13 to **13**-15
phase-shift, **3**-77, **3**-78
smoothing, **3**-59, **3**-66, **3**-73
test, **3**-75, **3**-77, **3**-78
Output (*see* Input and output)
Output impedance of operational amplifier, **2**-17

Output scanner, **4**-12 to **4**-14
Output stage of d-c amplifier, **2**-43, **2**-44
Oven, **2**-59, **2**-60, **4**-41
Overload hold, **4**-19, **4**-20
Overload indicator, **4**-11, **4**-19, **4**-20

PACE (*see* Electronic Associates, Inc.)
Pacific Electro-Nuclear Co., pulse-height analyzer, **19**-12, **19**-13
Packard-Bell Multiverter M-2, **7**-18
Padding for potentiometer linearity, **3**-6
Padé approximation of time delay, **6**-5 to **6**-7
Page memory, **20**-30 to **20**-33
Page writer, magnetic, **18**-19
PAM-PWM multiplier (*see* Time-division multiplier)
Paper, carbon, **18**-51
electrosensitive, **18**-51, **18**-52
resistance, **9**-80, **9**-81
wax, **18**-53
Parallel arithmetic units, **15**-17 to **15**-21, **16**-20 to **16**-27, **19**-26
timing and synchronization, **16**-20, **16**-21
Parallel circuits, **10**-6
Parallel correlator, **5**-95 to **5**-97
Parallel-feedback operational amplifiers, **2**-8, **2**-16, **2**-17
Parallel representation, **16**-13, **16**-14
Parameter-influence coefficients, **5**-153, **5**-154
Parameter optimization, analog techniques, **5**-71, **5**-151 to **5**-154
Parameters, ALGOL 60, **17**-30, **17**-31
Parametric oscillators, **11**-37, **11**-38, **13**-13 to **13**-15
Parasitic impedances in analog computers, **2**-20, **2**-58 to **2**-66, **4**-31, **9**-34, **9**-37 to **9**-45
Parenthesis-free notation, **19**-7
Parity-check alphabets, **14**-7, **14**-8
Partial differential equations, analog solution, **5**-110 to **5**-132
digital solution, **19**-4, **19**-5, **21**-8 to **21**-10
elliptic, **21**-10
hyperbolic, **21**-9, **21**-10
parabolic, **21**-8, **21**-9
Partial system tests, analog computers, **5**-64 to **5**-71
Patchbays, **4**-26 to **4**-30, **4**-35 to **4**-38, **6**-39
Patchboard, layout, **4**-26, **4**-29, **4**-30
leakage, **4**-27 to **4**-30
Patchboard verifier, **4**-21 to **4**-23
Payroll, **21**-14, **21**-15
Pens in plotters, **18**-51
Pentode amplifier, **11**-19
Pentodes, **10**-12, **10**-13
Per-unit quantities for network analyzer, **9**-63
Perforated tape (*see* Tape, punched paper)
Periodicity of information flow, **16**-14, **16**-15
Personnel for analog computers, **4**-41 to **4**-44
Phase inverter, block-diagram symbol, **1**-7
Phase-modulation recording, **12**-30, **12**-31
Phase shift in potentiometers, **6**-33
Phase-shift oscillator, **3**-77, **3**-78
Philbrick, G. A., Researches, Inc., integrators, **6**-23
Model K3-J, **6**-24
Model K-5M, **3**-45
operational amplifiers, K2-X, **6**-20
USA-3, **6**-21
special repetitive-computer components, **6**-18
Phillips Petroleum Company, **6**-14
Photoformer, **3**-68 to **3**-70, **3**-73, **3**-74
Photographic recording, **18**-53, **18**-57, **18**-58
Phugoid oscillation, **5**-58, **5**-59
Physical simulation, analog, **5**-64 to **5**-71
digital, **21**-56 to **21**-59
Physics problems, analog solution of, **5**-23 to **5**-34
Pile simulators, **5**-87 to **5**-93
Ping-pong register, **12**-13, **12**-14
Plane-stress analogy, **9**-20 to **9**-23
Plate analogy, **9**-23 to **9**-25
Plotkin alphabets, **14**-12
Plotters, **18**-50 to **18**-58
cathode-ray techniques, **18**-57, **18**-58
displacement control, **18**-53 to **18**-57
writing means, **18**-50 to **18**-53
Plug-in computing networks, **4**-35 to **4**-38
Plug-in potentiometers, **4**-37
Plugboards, **16**-2 to **16**-5
Pneumatic relay, simulation of, **5**-72
Poison generation in nuclear reactors, **5**-90, **5**-91
Poisson's equation, analog solution, **9**-76, **9**-77, **9**-81, **9**-82, **9**-84 to **9**-86
POLAR mode of resolver, **3**-23, **3**-26
(*See also* Resolvers)
Polish notation, **19**-7, **19**-8
Polynomial, evaluation of, **17**-7, **17**-8
Polynomial approximation, Chebyshev, **21**-47, **21**-48
least-square, **21**-45 to **21**-47
by ratio, **21**-48, **21**-49
Polynominal-equation solver (*see* Equation solvers)
Polynomial root extraction, analog techniques, **5**-139 to **5**-141
digital techniques, **19**-9, **19**-10
Polystyrene capacitor, **2**-55, **2**-61, **2**-62, **9**-44
Portland General Electric Co., **9**-61
Position-controlled flight table, **5**-69
Position-to-number converters, **18**-35 to **18**-41
POT SET mode, **4**-8
Potential, across condenser, **10**-5
across inductance, **10**-5
Potentiometer cleaner, **4**-55, **4**-56
Potentiometer linearity tester, **4**-55, **4**-56
Potentiometer loading, **2**-4, **2**-5, **3**-7 to **3**-9, **3**-23, **3**-38
for function generation, **3**-32 to **3**-35
(*See also* Load-equalizing resistors)

Potentiometers, coefficient, block-diagram symbol, 1-7
frequency response, 2-4, 2-64
helical, 2-3
equivalent circuit, 2-4, 2-64 to 2-66
high-frequency compensation, 2-4, 2-64 to 2-67
linearity, 2-3, 2-4, 3-4 to 3-8
specifications, 2-3 to 2-5
testing, 4-53
three-terminal, 2-8
comparison, 4-38
digital, 18-43, 18-44
servos, 18-32, 18-33
servo-driven, block-diagram symbols, 1-9, 1-10, 5-8 to 5-10
design and specifications, 3-4 to 3-9, 3-23 to 3-25, 3-32 to 3-39
sine-cosine, 3-23 to 3-25
tapped, 3-36 to 3-39
Power controls for analog computer, 4-3, 4-4 (*See also* Control systems)
Power level in network analog computers, 9-38 to 9-40
Power reactor, simulation of, 5-93
Power spectral density, 5-95
measurement of, 5-102 to 5-107
Power supply, analog computers, 2-80 to 2-82, 4-35
circuits, 2-72 to 2-82, 4-3, 4-4
reference, 2-81, 2-82, 4-38, 4-46
regulated, 2-80 to 2-82
small and medium-sized, 4-35, 4-38
Power-system network analyzers, 9-61 to 9-68
Precession, Bendix G-15, 20-17 to 20-19
Precision limiter, 3-62, 3-67, 3-68, 3-73
Preloading of potentiometers, 3-23
Pressure transmitter, simulation of, 5-73
Preventive maintenance of analog computers, 4-56, 4-57
Prewhitening, 5-96
Printed output, digital differential analyzer, 19-58, 19-59
Printers, Charactron, 18-7
line, 18-6 to 18-8
"on-the-fly," 18-6, 18-7
wheel, 18-6
wire, 18-6, 18-7
Probability density of random process, 5-94
Probability-density analyzers, 5-107, 5-108
Probability-distribution analyzers, 5-106, 5-107
Probe, for conductive-sheet analogy, 9-80
for electrolytic tank, 9-83, 9-84
Problem checking, analog computers, 4-18 to 4-26, 4-41, 5-63
excessive and insufficient signals, 4-19 to 4-21
programming checks, 4-21 to 4-26, 5-19, 5-20
Process control, analog computers, 5-71 to 5-86
chemical reactions, 5-82 to 5-87
linearized representation, 5-72 to 5-77
Process control, analog computers, nuclear reactor, 5-87 to 5-93
process models, 5-71, 5-78 to 5-80
digital computers, 21-84 to 21-86
(*See also* Control systems)
Program control, digital computers, 16-2, 16-8 to 16-11
general-purpose computers, 20-4, 20-20
input and output, 16-5 to 16-7
manual, 16-7
plugboard, 16-2 to 16-5
sequencing, 16-28 to 16-32
Program-logic command list, 17-14, 17-15
Programmer, input-output, 4-13, 4-15
Programming, analog computers, 5-5 to 5-22
automatic, 6-30 to 6-34, 9-67, 17-25, 17-26
analog storage in, 4-17, 4-18
programming equipment, 4-12 to 4-18
checking procedure, 4-21 to 4-26, 5-19, 5-20
linear, 5-149 to 5-151
records and summary, 5-20 to 5-22
sampled-data techniques, 6-31 to 6-34
semiautomatic, 4-12 to 4-18, 9-67
setup sheets, 5-19
cost of, 21-41
defined, 17-2
digital computers, 17-1 to 17-38
ALGOL 60, 17-26 to 17-37
example, 17-7, 17-8
flow diagrams, 17-8 to 17-10
sequencing, 16-2 to 16-5, 16-28 to 16-32
structure of programs, 17-8
digital differential analyzer, 19-61 to 19-67
hybrid analog-digital computers, 6-30
(*See also* Coding)
Proportional band, 5-74, 5-75
Proportional controllers (*see* Controllers)
Propositional calculus, 14-13, 14-14
Proton synchrotron design testing, 21-10, 21-11
Pulse-broadening circuits, 11-11 to 11-13
Pulse designations, 15-6
Pulse-generating circuits, 11-13
Pulse-height analyzers, 19-12, 19-13
Pulse recording, 12-30
Pulse train generator, 11-9
Pulse transformers, 10-16, 10-17
Pulse-width modulation in tape recording, 6-12, 6-13
Pulsed-attenuator multiplier (*see* Time-division multiplier)
Punched cards (*see* Cards)
Punched paper tape (*see* Tape)

Q of inductor, 9-42 to 9-44
Quadrant switching in resolvers, 3-29 to 3-31
electronic, 3-73

Quadratic equation, digital solution of, 17-15, 17-16
Quadratron, 3-75 to 3-77
Quadrature rejection, 8-22
Quantizers, 21-66, 21-67
Quarter-square multiplier, 3-42, 3-43, 3-76
Quartz memory line, 12-16, 12-17

Ramp compare, 18-31
Random-access storage unit, 12-90, 12-92 to 12-95
Random noise, digital simulation of, 21-59, 21-60
Random-noise generators, 3-81 to 3-83
(*See also* Noise generators)
Random process, 5-94
Random-process studies, analog computer, 5-94 to 5-109
repetitive, 6-31, 6-32
Random search for optimum parameters, 5-152, 5-153
Random telegraph wave, 3-81
Rate-aided hold circuit, 6-26 to 6-28
Rate resolver, 3-29 to 3-33
Rate time constant, 5-74, 5-75
RC circuits, 10-6, 10-7
RC coupled amplifiers, 11-25 to 11-33
RC networks, 2-8 to 2-15
RCA Laboratories Division, 3-54
REAC[R] 400-series computers, 4-33
(*See also* Reeves Instrument Corporation)
Reaction rate, chemical, 5-83
Reactors, chemical, 5-82, 5-83
nuclear, simulation of, 5-87 to 5-93
Read amplifier circuit, 12-88 to 12-90
Read-out, nondestructive, 12-103 to 12-105
Realizability of analogy, 9-7
Recirculating memory, 16-12, 16-13
Recorder calibration, 4-15, 4-16
RECTANGULAR mode, 3-23
Redundancy defined, 14-3, 14-4
Redundant codes, 14-4, 14-5
Reed-Muller alphabets, 14-11, 14-12
Reeves Instrument Corporation, Model A-400-1, 2-73, 2-79
Project Cyclone, 4-43
REFERENCE OFF mode, 4-4, 4-8, 4-9
Reference power supplies, 2-81, 2-82, 4-38
checking of, 4-46
Regenerative integrator connections, 8-11
Register, 16-12, 16-14, 16-18 to 16-25, 17-11, 17-12
and accumulator, 15-10, 15-11
arithmetic, 15-10 to 15-21, 16-17 to 16-27, 20-11, 20-13, 20-17
circulating, 15-10, 16-12, 16-18, 19-18, 20-8
digital differential analyzer, 19-59 to 19-61
index, 16-11, 17-11, 17-12, 20-27 to 20-29
shift (*see* Shift register)
Regulated power supply, example, 2-80 to 2-82
Regulator amplifiers, 11-24, 11-25
Relaxation, analogy for, 9-86, 9-87
Relaxation method, 5-144, 5-157
Relay comparator (*see* Comparator)
Relay computers, 20-2 to 20-5
arithmetic unit, 20-4, 20-5
characteristics, 20-3 to 20-5
code, 20-3
control, 20-4
input devices, 20-5
output devices, 20-5
storage, 20-3, 20-4
Relay drivers, 11-7
Relay timing test, 4-49
Relays, analog computer, 3-63
applications, 5-40
simulation of, 5-40, 5-41
testing, 4-53, 4-54
Remote control of analog computers, 4-11
(*See also* Slave operation)
Repeat command, 17-24
Repetitive analog computers, 6-17 to 6-25
sampled-data techniques, 6-31 to 6-34
vs. "slow" computers, 6-17
special components, 6-18 to 6-20
REPETITIVE mode, 4-9
Reset circuits, 6-22 to 6-24
RESET mode, 4-4, 4-5, 6-32 to 6-34
of continuous resolver, 3-30
Reset rate, 5-74, 5-75
Resistance-coupled amplifier tables, for d-c amplifier design, 2-47, 2-48
Resistance-network analogy, 9-69 to 9-80
Resistance paper, 9-80, 9-81
Resistance switches, 13-7 to 13-9
Resistors, for analog computers, cathode-bias, effect on d-c amplifier, 2-45, 2-46
computing, 2-58, 2-59
for electronic computers, 2-58, 2-59
for network computers, 9-37, 9-39, 9-44, 9-49, 9-62
screen series, effect on d-c amplifier, 2-45, 2-46
testing, 4-46 to 4-49
for digital computers, 10-8 to 10-10
nonlinear, 13-8, 13-9
switch, 13-7
Resolution of potentiometer, 2-3
Resolver phase-shift converter, 18-37, 18-38
Resolver scale factor, 3-23
Resolvers, continuous, 3-29 to 3-31, 3-61, 3-73
in direct-analog computers, 9-58
electronic, 3-56 to 3-61, 3-73, 7-26 to 7-28
inverse, 1-10, 3-23, 3-25 to 3-28, 3-59
servo, 1-10, 3-4, 3-19 to 3-33, 5-55, 8-16, 8-17, 8-21
d-c, 3-23 to 3-25
induction, 3-19 to 3-23
testing, 4-52
Ring analogy, 9-26 to 9-29
Ringing circuits, 11-8, 11-9
Ripple in time-division multipliers, 3-54
Ripple filters for multipliers, 3-52, 3-53
RLC circuits, 10-8

Roll-off networks, **2**-49 to **2**-51
Roll table, **5**-65, **5**-66
Root extraction, analog, **3**-76, **5**-7, **5**-12, **7**-25, **7**-26
  polynomial equations, **5**-139 to **5**-141
  digital, high-degree polynomials, **19**-9, **19**-10
Root-locus tracking, **5**-146 to **5**-148
Root-mean-square adder, **3**-76, **7**-25, **7**-26, **8**-21
Rotating-head tape, **18**-18, **18**-19
Round-off error, **19**-46 to **19**-48
Royal-McBee LGP-30, **20**-12 to **20**-16
Rubber-sheet analogy, **9**-84, **9**-85
Rubber sheets, **9**-81
Runge-Kutta method of equation solving, **6**-43, **21**-51, **21**-54
  modified, **21**-4, **21**-5
Russell beam analogy, **9**-15 to **9**-17

SADSAC, **6**-29, **6**-30
Sample-hold approximation of time delay, **6**-8
Sample-hold circuit, **5**-37, **6**-26, **6**-27, **6**-31, **6**-44
  (*See also* Storage, analog)
Sample-hold read-out for repetitive analog computer, **6**-31
Sampled data, in analog-digital systems, **6**-43 to **6**-48
  in repetitive analog computers, **6**-31 to **6**-34
Sampled-data system, simulation of, **5**-37, **5**-42
Sampling multiplier, **3**-53
Sampling-type noise generator, **3**-81, **3**-83
Sandbed mapper, **9**-85, **9**-86
Scalar product, **17**-17 to **17**-20, **17**-23
  cyclic code, **17**-18
  generator for, **17**-20
  without index registers, **17**-18, **17**-19
  with linear code, **17**-19
  using subroutines, **17**-23
Scale change, automatic, **4**-17, **4**-18, **5**-61, **5**-62
Scale factors, calculation, **5**-13
Scaling, analog computers, direct analogy, **9**-5, **9**-6
  electronic, **5**-12 to **5**-19
  mechanical, **8**-5, **8**-11
  network analyzer, **9**-63
  resolvers, **3**-23
  digital differential analyzer, **19**-31, **19**-32, **19**-65 to **19**-67
Scanners, cathode-ray-tube, **3**-69, **3**-70
  output, **4**-12 to **4**-14
Scanning, analog computer read-out, **4**-12, **4**-14
Scanning-type equation solver, **5**-139 to **5**-144
Schmidt trigger, **11**-11
Scintillation noise generator, **3**-81
Scotch yoke, **3**-23, **8**-7
Screen-grid resistor, **11**-19
Screen resistors, effect on d-c amplifier, **2**-45, **2**-46
SEAC, **21**-2
Semibell function, **5**-19
Semiconductor choppers, **7**-2 to **7**-6, **7**-32 to **7**-34
Semiconductor diodes (*see* Diode)
Semielastic stop (*see* Limit stop)
Sense windings, **12**-60 to **12**-64
Separation of variables, **5**-112, **5**-119, **5**-123
Separator symbols, **17**-3
Sequencing, digital computers, **16**-2 to **16**-4, **16**-28 to **16**-32
  arithmetic operation, **20**-23, **20**-24
  asynchronous, **16**-32
  one-address, **20**-21
  stored-program, **16**-29, **16**-30
  synchronous, **16**-30 to **16**-32
  wired-plugboard, **16**-28, **16**-29
Sequential computation, automatic, **6**-32 to **6**-34
Serial arithmetic units, **15**-8 to **15**-16
  examples described, **20**-11, **20**-14, **20**-17
  timing and synchronization, **16**-18 to **16**-20
Serial binary accumulator, **15**-12, **15**-13, **16**-13
Serial binary adder, **15**-9, **15**-10
Serial correlator, **5**-95 to **5**-97
Serial half-adder, **15**-8, **15**-9
Serial register (*see* Register)
Serial representation, **15**-8, **16**-13, **16**-14
Series circuits, **10**-6
Series limiter, **3**-62 to **3**-65
  dual, **3**-66
Servo amplifiers, **3**-9 to **3**-11
Servo motors, **3**-11 to **3**-13
Servo plotting table as function generator, **3**-79
Servo recording tables, **6**-16
Servo resolvers (*see* Resolvers)
Servo-set coefficient potentiometers, **2**-6, **2**-7
Servomechanisms, analog representation of, **5**-35 to **5**-42
  bandwidth of, **3**-3, **3**-17
  defined, **5**-35
  digital differential analyzer, **19**-35 to **19**-39
  digital potentiometer, **18**-32, **18**-33
  linear, **5**-35
  nonlinear, **5**-35, **5**-37
  sampled-data systems, **5**-37, **5**-42
Servomultipliers, **3**-4 to **3**-19, **4**-35
  block-diagram symbols, **1**-9, **5**-8, **5**-9
  division with, **5**-8 to **5**-11
  multiplication with, **5**-5, **5**-8, **5**-9
  testing of, **4**-50 to **4**-53
Set point, **5**-71
SETUN, **15**-4
Setup sheets, **5**-19
Shaping filter, **5**-96, **5**-100
Shell analogy, **9**-29, **9**-30
Shielded patchbay, **4**-26 to **4**-29
Shift-jump command, **17**-24, **17**-25
Shift operations, **20**-9, **20**-10

Shift register, a-c vs. d-c coupling, **16**-24, **16**-25
  double, **16**-21
  dynamic primary storage, **16**-23, **16**-24
  ferroelectric, **12**-115, **12**-116
  ferromagnetic, **12**-5 to **12**-16
  flip-flop, **16**-14
  magnetic, **12**-13, **12**-14
  serial, **16**-18
  single, **16**-22, **16**-23
Short-circuit transfer impedance, **2**-20
Shunt limiter, **3**-62, **3**-64
Sift command, **17**-25
Signal generator, **3**-75, **3**-77, **3**-78
Signal-modifying circuits, digital computer, **11**-7 to **11**-13
Signum function, **19**-39
Similar-triangle multiplier, **8**-5
Simulag Group Model 1.002, **6**-12, **6**-13
Simulag Group Model 1.011, Magnetic Drum, **6**-14, **6**-16
Simulation, analog computers, dynamical-system studies, **5**-42 to **5**-49
  physical (partial system tests), **5**-64 to **5**-71
  process-control system, **5**-71 to **5**-83
  of servomechanisms, **5**-35 to **5**-42
  combined analog and digital, **6**-42 to **6**-52, **21**-54 to **21**-56
  defined, **21**-42
  digital computers, **21**-42 to **21**-62
  airplane flight, **21**-56 to **21**-59
  dynamic behavior of simulator, **21**-49 to **21**-54
  electric networks, **21**-60, **21**-61
  human behavior, **21**-61, **21**-62
  random noise, **21**-59, **21**-60
  stages of, **21**-43
Simulation board, analog computer, **4**-36 to **4**-38
Simulators, digital, **21**-42, **21**-43, **21**-49
Simultaneous-equation solvers, **5**-133 to **5**-138, **19**-2, **19**-3
Sine-cosine function generator, **7**-23
  (*See also* Resolvers)
Sine-cosine potentiometers, **3**-23 to **3**-25
  block-diagram symbol, **1**-10
Sine-generating loop, **2**-60, **2**-70 to **2**-72
Sine-wave damping in analog computers, **2**-70 to **2**-72
Single-address computer (*see* One-address computer)
Single's approximation of time delay, **6**-5
Six-diode switch, **6**-26, **6**-28
Slave operation, of analog computers, **4**-11
  of electronic multipliers, **1**-8, **3**-51, **5**-8, **5**-9, **7**-11
Slave variable, **3**-3
Slicer circuits, **5**-106 to **5**-109
Slip error, **8**-2
SLO-RESET mode, **4**-9
Smoothing oscillator, **3**-59, **3**-66, **3**-73
Soap-film analogy, **9**-85
Solenoid-ratchet motors, **18**-46, **18**-47
Sorting, **21**-22 to **21**-33
  choice of techniques, **21**-32, **21**-33
  digital, **21**-24 to **21**-26, **21**-28, **21**-29, **21**-32, **21**-33
  edge-punched cards, **21**-26, **21**-27
  electromechanical, **21**-27, **21**-28
  internal, **21**-24, **21**-31 to **21**-33
  merge, **21**-24, **21**-29 to **21**-33
  needle, **21**-26, **21**-27
  punched cards, **21**-27, **21**-28
  using magnetic tapes, **21**-28, **21**-29
Source destination, stored-program computers, **20**-5, **20**-6
Source impedance of operational amplifier, **2**-17
Space Technology Laboratories, Inc., **4**-43, **4**-58
Spare parts of analog computers, **4**-58
Special-purpose (buffer) storage units, **12**-91 to **12**-97
Special-purpose computers (*see* Digital computers, special-purpose)
Spectral densities, **5**-95
  measurement of, **5**-102 to **5**-107
Spectrum analyzers, **5**-96, **5**-102 to **5**-107
Spline interpolation, **21**-71, **21**-72
Square-card sine-cosine potentiometer, **3**-24
Square-root extraction (*see* Root extraction)
Square-root generation, analog, **3**-76, **5**-7, **5**-12, **7**-25, **7**-26
Stability, of aircraft (*see* Flight equations)
  of computer setups, **2**-58
  of operational amplifier, **2**-41
Stability chart, equation solution, **21**-52 to **21**-54
Stabilization of computer servos, **3**-14, **3**-15
Stabilization networks, **2**-49 to **2**-51
  external, **2**-51
Stanford Research Institute, **21**-36
Statements, **17**-5 to **17**-7, **17**-27, **17**-28
  array, **17**-5
  assignment, **17**-5
  conditional, **17**-6
  DO, **17**-5, **17**-6
  GO TO, **17**-6
  grouping of, **17**-7
  one-address computer, **20**-20, **20**-21
  syntax of, **17**-34, **17**-35
  type, **17**-5
Static accuracy of computer servo, **3**-16
Static check of analog computation, **4**-23
Static current generator, **9**-54
Static error coefficients, **5**-37
Static friction, simulation of, **5**-40, **5**-45
Static primary storage, **16**-21 to **16**-23
Stationary random process, **5**-94
Steady-state displacement error, **5**-37, **5**-39
Steepest-descent methods, **3**-59, **5**-135 to **5**-137, **5**-140, **5**-145 to **5**-147, **5**-152
Step motors, **18**-46, **18**-47, **21**-65
  servos, **18**-48
Step multiplier, **7**-15 to **7**-18
Stepping devices, digital-to-analog conversion, **21**-65

Stepping relays in analog computations, **6**-29, **6**-30
Stochastic processes, **21**-11
Stop, simulation of (*see* Limit stops)
Storage, analog, **5**-48, **5**-153, **6**-26 to **6**-34
  capacitor, time delay, **6**-8 to **6**-10
  circuits and applications, **4**-17, **6**-26 to **6**-29
  solution of equations, **6**-52 to **6**-54
 digital, delay-line primary, **16**-24
  dynamic (regenerated) primary, **16**-23, **16**-24
  machine-tool control, **21**-77 to **21**-79
  magnetic-core, **12**-41 to **12**-106
    buffer, **12**-91 to **12**-97
    coincident-current, **12**-80 to **12**-86, **12**-98 to **12**-100, **12**-102, **12**-103
    random-access, **12**-90, **12**-92 to **12**-94
  relay type, **20**-3, **20**-4
  static primary, **16**-21 to **16**-23
 digital differential analyzer, **19**-40 to **19**-42
 (*See also* Memory)
Stored-carry multiplier, **15**-18, **15**-19
Stored overload, **4**-19, **4**-20
Stored program, defined, **16**-2
 plugboard control, **16**-5
Stored-program computers, address modification, **20**-28, **20**-29
 Bendix G-15, **20**-15 to **20**-19
 classification, **20**-5 to **20**-7
 defined, **16**-2
 four-address, **16**-10, **16**-11, **17**-24, **20**-26, **20**-27
 IBM 650, **20**-10 to **20**-12
 index register, **16**-11, **17**-11, **17**-12, **20**-27 to **20**-29
 interrupt feature, **20**-30 to **20**-33
 one-address, **16**-7, **16**-8, **16**-11, **17**-23, **20**-20 to **20**-23
 page memory, **20**-30 to **20**-33
 Royal-McBee LGP-30, **20**-12 to **20**-16
 structure, **16**-7, **16**-8, **20**-19
 three-address, **16**-10, **17**-23, **17**-24
 two-address, **16**-9, **17**-23, **20**-24, **20**-25
Strobed helix-bar recorder, **18**-56, **18**-57
Strobed indicators, **18**-22, **18**-23
Stromberg-Carlson 5000 printer, **18**-7
Structural analogies, direct-analogy computers, dynamic, **9**-12 to **9**-19
  static, **9**-19 to **9**-30
Subroutine, **17**-8, **17**-11
 example, **17**-15, **17**-16
 MARK TRANSFER in organization, **17**-22
 scalar product using, **17**-23
Subtraction, analog, mechanical, **8**-2 to **8**-4
 digital, **14**-3, **15**-12
 digital differential analyzer, **19**-39, **19**-40
 selective, in voltage-to-number converters, **18**-32, **18**-33
Summarize command, **17**-25
Summing amplifier, block-diagram symbol, **1**-7
 capacitive compensation, **2**-67 to **2**-69
 design, **2**-42
Superconductive switches, **13**-13
Superposition integral, **5**-95
Surge tank, simulation of, **5**-78
SWAC, **20**-26, **21**-2
Sweep-count devices, **18**-37, **18**-38
Switch (programming), **17**-8, **17**-34
Switches, analog computer, electronic, **6**-26 to **6**-28
  network, synchronous switch, **9**-51 to **9**-54
 digital computer, bidirectional, **12**-77, **12**-78
  binary, **12**-68 to **12**-70
  coincident-current drive systems, **12**-83 to **12**-86
  diode, **12**-79, **13**-9, **14**-20
  double-coincidence and triple-coincidence, **12**-69, **12**-70
  exponential, **14**-20, **14**-21
  ferroelectric, **12**-114, **12**-115, **13**-11 to **13**-13
  magnetic, **12**-8, **12**-9, **12**-64 to **12**-80, **13**-10
  multiplicative, **14**-20, **14**-21
  parametric amplifier, **13**-13 to **13**-15
  resistance, **13**-7 to **13**-9
  superconductive, **13**-13
  unidirectional, **12**-71 to **12**-77
 transistor (*see* Transistor switches)
Switching, electronic, of analog voltages, **3**-50, **3**-69, **3**-73, **6**-26 to **6**-29
  of integrators, **6**-22 to **6**-25, **6**-27, **6**-28
Switching theory, **14**-13 to **14**-23
Symbol displays, **18**-20 to **18**-29
 cathode-ray methods, **18**-25 to **18**-29
 disks, **18**-22
 drums, **18**-21
 illuminated symbols, **18**-22 to **18**-25
  deflection-formed, **18**-27, **18**-28
  dot-formed, **18**-25, **18**-27
  edge-lit, **18**-23, **18**-24
  projector display, **18**-24
 pointers, **18**-22
 positioned symbols, **18**-21, **18**-22
 for repetitive analog computers, **6**-24, **6**-25
 tapes, **18**-21, **18**-22
Symbol stamps, **18**-50, **18**-51
Symbols, coding, **17**-2, **17**-3, **17**-36, **17**-37
  ALGOL 60, **17**-26 to **17**-37
    basic classification, **17**-36, **17**-37
  grouping, **17**-3
  operational, **17**-2, **17**-3
  separators, **17**-3
 design of, **18**-20, **18**-21
 displays (*see* Symbol displays)
Symbols (notation), for analog computers, **1**-6 to **1**-10, **5**-5, **8**-4
 for digital computers, binary coding circuits, **15**-5, **15**-6
  switch wiring diagrams, **12**-67
 (*See also* Block diagrams and symbols)
Synchronization, in analog computer servos, **3**-19
 in analog-to-digital conversion, **6**-45

Synchronization, of digital computers, **16**-14 to **16**-21
  control section, **16**-28 to **16**-33
Synchronous motors, **18**-47, **18**-48
Synchronous switches for network analog computers, **9**-51 to **9**-54
Syntax, ALGOL 60, of declarations, **17**-35, **17**-36
  of expressions, **17**-31 to **17**-34
  of statements, **17**-34, **17**-35
System design, analog computers, large installations, **4**-40, **4**-41
  small and medium-sized, **4**-35 to **4**-39
 digital computers, **16**-2 to **16**-33
 management problems, **21**-36, **21**-37
Systron-Donner Corp. (*see* Donner Scientific Company)

Tachometer, specifications, **3**-31
Tachometer-controlled flight table, **5**-68
Tachometer feedback, **3**-15 to **3**-19
Tangent circuits, **3**-34
Tangent tracer, **8**-5
Tap resistance, **3**-8
Tape, analog, **6**-10, **6**-13 to **6**-16, **18**-13, **18**-14
 digital, **6**-13, **6**-14, **12**-18, **12**-19, **12**-32, **12**-33, **18**-1 to **18**-5
  materials, **12**-43, **12**-44, **18**-2
 digital differential analyzer, **19**-55 to **19**-59
 files, multitape, **21**-19, **21**-20
 magnetic, **6**-10 to **6**-16, **12**-18, **12**-32, **12**-33, **18**-13 to **18**-20
  converters from, **18**-15 to **18**-18
  converters to, **18**-11, **18**-15
  digital sorting with, **21**-28, **21**-29
  rotating-head system, **18**-18, **18**-19
  signals, **18**-18
 punched paper, **6**-14, **18**-1 to **18**-5
  punching, **18**-5
  reading, **18**-4, **18**-5
 symbol display, **18**-21, **18**-22
 (*See also* Magnetic recording)
Tape loop for analog data, **6**-10
Tapped potentiometers, calibration, **3**-36, **3**-38, **3**-39
 as function generators, **3**-36 to **3**-39
 for functions of two variables, **3**-79
Taylor-series desampler, **6**-47, **6**-48
Taylor-series expansion, **5**-122
Teflon capacitor, **2**-61, **2**-62
Teledeltos, **9**-80, **9**-81
Telephone systems, automatic, **19**-13, **19**-14
Teletype codes, **18**-2, **18**-3
Teletype punch, **18**-5
Teletype reader, **18**-4
Terminal linearity, **3**-5
Ternary number system, **15**-4
Test boards, **4**-56
Test circuits, analog computers, **4**-44 to **4**-54
Test equipment, analog-computer laboratory, **4**-54 to **4**-57
Test oscillator, **3**-75, **3**-77, **3**-78
Thermocouple, simulation of, **5**-72
Three-address computer, **16**-10
 command system, **17**-23, **17**-24
Three-beam circuit, **9**-18, **9**-19
Time average, **5**-95
 sampled-data computation, **6**-31, **6**-32
Time constant, **10**-7
Time-correlation functions, **5**-95, **5**-98
Time delay, in analog-to-digital conversion, **6**-48, **6**-49
 analog representation, **6**-3 to **6**-16
  capacitor-storage approximations, **6**-8 to **6**-10, **6**-33
  magnetic recording schemes, **6**-14 to **6**-16
  operational-amplifier approximations, **6**-5 to **6**-7
  survey of analog circuits, **6**-4
  use of servo tables, **6**-16
Time-division multiplier, **3**-50 to **3**-57, **7**-9 to **7**-14
Time-to-number converters, **18**-41
Time scale, **5**-17, **5**-18
Time-scale check, **4**-24
Time sharing, analog computing elements, **6**-27, **6**-28
Timer, analog circuit, **3**-71
Timing of digital computers, **16**-14 to **16**-21, **20**-8, **20**-9
 carry operation, **16**-25 to **16**-27
 control section, **16**-28 to **16**-33
 Royal-McBee LGP-30, **20**-14
Topological similarity, **9**-5
Torsion of beams (*see* Beam equations)
Torsion rod, analogy of, **9**-12
Total error, **19**-48 to **19**-50
"Totem-pole" amplifier, **2**-44
TRACK mode (*see* RESET mode)
Tracking accuracy of computer servo, **3**-16
Tracking-type equation solver, **5**-139, **5**-140, **5**-144 to **5**-149
Trailers, **21**-19 to **21**-21
Trainers, flight, **5**-65, **21**-56 to **21**-59
Transducers, heat-transfer, **8**-18 to **8**-20
  self-heating, **8**-20 to **8**-22
Transfer, of control, **17**-11, **20**-4
  one-address computer, **20**-24
 of information, **16**-16, **16**-17
  access time and, **16**-15, **16**-16
  between memories, **16**-16, **16**-17
 subroutine, **20**-23
Transfer admittance, **9**-45, **9**-46
Transfer functions, differential-analyzer representation, **2**-28 to **2**-40
 of networks, **2**-10 to **2**-15
 of operation amplifiers, **2**-16, **2**-18
Transfer impedances, **9**-45, **9**-46
 for operational amplifiers, **2**-20 to **2**-27
Transfer loops in ferromagnetic shift registers, **12**-7 to **12**-10
Transformer(s), coupling, **11**-34
 magnetic-core drive, **12**-82, **12**-83
 for network analog computers, **9**-4, **9**-34, **9**-39, **9**-41 to **9**-44, **9**-63
 pulse, **10**-16, **10**-17

Transformer-coupled amplifiers, **11**-33, **11**-34
Transformer-coupled driver, **11**-36, **11**-37, **12**-82, **12**-83
Transformer switches, **13**-10
Transistor(s), **10**-29 to **10**-56
  alloy junction, **10**-48
  "alpha less than unity," **10**-48
  avalanche, **10**-51
  avalanche multiplication, **10**-32
  characteristics, **10**-52
  collection factor α, **10**-31
  diffused junction, **10**-49
  double-base diode, **10**-51
  electrical ratings, **10**-47
  field effect, **10**-50
  frequency cutoff, **10**-31
  hook collector, **10**-51
  intrinsic barrier, **10**-50
  junction capacitance, **10**-31
  large-signal static characterization, **10**-33 to **10**-38
  large-signal transient characterization, **10**-39 to **10**-41
  measurement and test, **10**-41
  mechanical ratings, **10**-48
  negative-resistance characteristics, **10**-37, **10**-38
  point contact, **10**-50
  power ratings, **10**-47
  reliability, **10**-54 to **10**-56
  reverse cutoff currents, **10**-30
  small-signal measurements, **10**-45, **10**-46
  small-signal transient characterization, **10**-38, **10**-39
  surface barrier, **10**-49
  tetrode, **10**-48
  transient measurements, **10**-45
  transport factor β, **10**-31
  types of, **10**-30
Transistor chopper (*see* Semiconductor choppers)
Transistor circuit, buffer inverter, **11**-6
  driver, **11**-7
  emitter follower, **11**-2 to **11**-4
  one-shot multivibrator, **11**-12
  Schmidt trigger, **11**-11
Transistor function generators, **7**-20 to **7**-29
  bidirectional limiters, **7**-20, **7**-21
  inverse-trigonometric-function, **7**-21 to **7**-23
  rectangular-to-polar coordinate conversion unit, **7**-26 to **7**-28
  square-root, **7**-25, **7**-26
  trigonometric-function, **7**-23 to **7**-25
Transistor limiter, **7**-15, **7**-20, **7**-21
Transistor multipliers, binary-increment, **7**-15 to **7**-18
  division with, **7**-18 to **7**-20
  Hall-effect, **7**-18
  magnetoresistor, **7**-18
  phase-amplitude, **7**-12 to **7**-14
  time-division, **7**-9 to **7**-14
  triangle-integrating, **7**-14, **7**-15
Transistor switches, analog, **6**-24, **6**-32 to **6**-34, **7**-6 to **7**-9
  digital, **12**-80, **12**-87
Transistorized analog computers, **7**-1 to **7**-36
  function-generator circuits, **7**-20 to **7**-29
  multiplier circuits, **7**-9 to **7**-20
  transistors vs. vacuum tubes, **7**-2, **7**-3
Translation circuits, **16**-5
Transmission loss in power system, **5**-155
Transonic-flow analogy, **9**-87
Transpolarizer, **12**-112, **12**-113
Transport delay (*see* Time delay)
Tree (program structure), **14**-20, **14**-21, **17**-8
Triangle-integration multiplier, **3**-42 to **3**-46, **7**-14, **7**-15
Trigonometric-function generators, transistor, **7**-23 to **7**-25
  inverse, **7**-21 to **7**-23
Triode amplifier, **2**-45
Triodes, **10**-11, **10**-12
Triple-coincidence matrices, **12**-54, **12**-55
Triple-coincidence switches, **12**-69, **12**-70
Truncation error, **19**-42 to **19**-46
Trunk, block-diagram symbol, **1**-10
Truth-table computer, **19**-6
Truth tables, **14**-14, **14**-15
Truth values, **14**-13
Tubes, display, conventional cathode-ray, **18**-26
  image-converter, **18**-28, **18**-29
  shaped-beam, **18**-29
  shaped-electrode, **18**-24
  Williams-tube system, **12**-40, **12**-41
  vacuum, **10**-10 to **10**-16
  beam power, **10**-14
  gates, **13**-1 to **13**-3
  (*See also* Cathode-ray tubes)
Two-address computer, **16**-9, **20**-24, **20**-25
  command system, **17**-23, **20**-25
Two-time-scale control system, **6**-33, **6**-35, **6**-41
Two's complementer, **15**-12

Unit records, **21**-19 to **21**-21
United States Air Force, **21**-40
  USAF-Fairchild computer, **19**-2 to **19**-4
Univac card-to-tape converter, **18**-11

Vacuum-tube voltmeter, **4**-10
Vacuum tubes (*see* Tubes)
Validity, testing for, **14**-15, **14**-16
Valve, simulation of, **5**-35, **5**-38
Valve actuator, simulation of, **5**-72, **5**-74
Van der Pol's equation, **5**-15 to **5**-17
Variables, machine, defined, **1**-4
  multiplication, digital differential analyzer, **19**-29 to **19**-35
  separation of, **5**-112, **5**-119, **5**-123
  subscripted, **17**-6, **17**-27, **17**-32
  substitution of, **19**-65

Varistor function generator, **3**-73, **3**-75 to **3**-77
Vector resolution (*see* Resolvers)
Velocity Coordinate System (VCS), **5**-50
Velocity-earth Coordinate Systems (VECS), **5**-51
Velocity error, **5**-37, **5**-39
Velocity servo, **3**-29
Vendac, **19**-7
Ventilation of cabinets, **4**-33, **4**-34
Vernier techniques, decimal, **18**-38, **18**-39
Vernier waveforms, **18**-38, **18**-42, **18**-49
Volt, **10**-4
Voltage-amplifier stages of d-c amplifier, **2**-43
Voltage-to-number converters, **18**-30 to **18**-35
Voltage scale (*see* Scale change; Scale factors)

Watt, **10**-5
Wave equation, analog solution, **9**-78, **9**-79
Weighting function, **6**-54, **6**-55
Western Union Telegraph Co., **9**-80
Westinghouse A-C Network Calculator, **9**-61 to **9**-63
Westinghouse EDC, **5**-157
Wheel, code, position-to-number converter, **18**-36, **18**-39, **18**-40
  servo, **18**-45, **18**-46, **18**-48
Wheel-and-disk integrator, **8**-3 to **8**-5
Wheel printer, **18**-6
Whiffletree multiplier, **15**-20
White noise, **5**-96, **5**-99 to **5**-101
Wien-bridge oscillator, **3**-78
Wiener-Khintchine relations, **5**-95, **5**-105
Wiener-Lee relation, **5**-95
Williams-tube memory, **12**-34 to **12**-41
  circle-dot display, **12**-38
  defocus-dash display, **12**-38
  defocus-focus display, **12**-38
  double-dot display, **12**-37
  operating factors, **12**-38 to **12**-41
  read-around ratio, **12**-39
Wind Axes, **5**-50
Windings, digit-plane, **12**-55, **12**-56
  inhibit, **12**-55, **12**-56
  linear or word-selection reading, **12**-58 to **12**-60
  sense, **12**-60 to **12**-64
Word, pulse designation, **15**-6
  representation, **16**-13
Word clock, **16**-14
Word-selection method of reading, **12**-58 to **12**-60
Word structure, Bendix G-15, **20**-15
  IBM 650, **20**-10
  Royal-McBee LGP-30, **20**-12
Writing in plotters, **18**-50 to **18**-53

Xenon poisoning, simulation of, **5**-90, **5**-91
Xerography, **18**-52, **18**-58

Z line, digital differential analyzer, **19**-22 to **19**-26
Zener diodes, in analog computer, **3**-67, **6**-27
  in coincidence gate with memory, **13**-7
  in flip-flop, **12**-4
Zero, representation of, **15**-23
Zero-based linearity, **3**-5, **3**-6
Zero control on plugboard, **16**-6
Zero-order hold (*see* Sample-hold circuit)